

# Pavages: du local au global

Mathieu Sablik  
LATP, Université de Provence  
sablik@latp.univ-mrs.fr

22 octobre 2012  
Version 1.0<sup>1</sup>

1. Cette première version contient certainement des coquilles, elle sera mise à jour et accessible à l'adresse suivante: <http://www.latp.univ-mrs.fr/~sablik/>

# Table des matières

<b>1</b>	<b>Notion de pavages</b>	<b>3</b>
1.1	Définition . . . . .	3
1.2	Groupe des isométries . . . . .	3
1.3	Ensemble de pavages . . . . .	7
1.4	Pavages par polygones réguliers . . . . .	8
1.5	Règles locales . . . . .	8
<b>2</b>	<b>Éléments de dynamique symbolique</b>	<b>11</b>
2.1	Sous-shift . . . . .	11
2.2	Sous-shift de type fini et sofiqes . . . . .	14
2.3	Quelques invariants topologiques . . . . .	18
<b>3</b>	<b>Pavages apériodiques</b>	<b>23</b>
3.1	Pavage de Robinson . . . . .	23
3.2	Sous-shift substitutifs . . . . .	28
3.3	Pavages substitutifs avec des polygones . . . . .	35
3.4	Pavages apériodique à 13 tuiles de Kari-Culik . . . . .	35
<b>4</b>	<b>Pavages et modèle de calcul</b>	<b>39</b>
4.1	Problèmes de décision . . . . .	39
4.2	Machines de Turing . . . . .	40
4.3	Problème de complétion . . . . .	43
4.4	Problème du pavage périodique . . . . .	44
4.5	Le problème du domino . . . . .	44
<b>5</b>	<b>Opérations dynamiques sur un sous-shift</b>	<b>53</b>
5.1	Opérations et simulations . . . . .	53
5.2	Un résultat de simulation . . . . .	57
5.3	Autres travaux . . . . .	68
<b>6</b>	<b>Quelques applications du théorème précédent</b>	<b>69</b>
6.1	Entropie atteignable par un sous-shift de type fini ou un sofique . . . . .	69
6.2	Un S-adic effectif multidimensionnel est sofique . . . . .	71
6.3	Pavages obtenus par coupé-projection réalisables par des sofiqes . . . . .	73
<b>7</b>	<b>Exercices</b>	<b>80</b>
7.1	Dynamique symbolique . . . . .	80
7.2	Pavage apériodique . . . . .	82
7.3	Problème de décisions . . . . .	82

# Chapitre 1

## Notion de pavages

### 1.1 Définition

#### 1.1.1 Tuiles et pavages

Une *tuile* est une sous ensemble de  $\mathbb{R}^d$  homéomorphe à une boule. Étant donné qu'un homéomorphisme préserve les propriétés topologiques, une tuile  $t$  a immédiatement les propriétés suivantes :

- $t$  est compact,
- l'intérieur de  $t$  et le complémentaire de  $t$  sont connexes,
- le bord de  $t$  est égal au bord de l'intérieur de  $t$ ,
- le bord de  $t$  est homéomorphe à la sphère de dimension  $d$ .

*Exemple 1.1.1.* Une tuile de  $\mathbb{R}$  est un segment. Dans  $\mathbb{R}^2$  on utilise généralement des polygones mais il peut apparaître des tuiles fractales.

Un *pavage* de  $\mathbb{R}^d$  est une famille de tuiles  $\mathbf{t} = (t_i)_{i \in \mathbb{N}}$  telle que

- tout point de  $\mathbb{R}^d$  appartient à une tuile,
- $t_i \cap t_j = \emptyset$  pour tout  $i \neq j$ .

*Remarque.* Quelques remarques sont à faire sur cette définition :

- une famille de tuiles qui vérifient les conditions précédentes est forcément infinie mais reste dénombrable ;
- les bords de deux tuiles distinctes peuvent s'intersecter ;
- on s'est placé dans le cadre de  $\mathbb{R}^d$  muni de la métrique euclidienne mais cette définition peut aussi s'adapter à d'autres espaces ;
- la définition est très générale car un pavage peut être défini par des tuiles de différentes formes.

### 1.2 Groupe des isométries

#### 1.2.1 Groupe des isométries d'un pavage

On rappelle que le groupe des isométries  $\mathcal{I}_d$  de  $\mathbb{R}^d$  est engendré par les isométries de type :

- translation de vecteur  $\mathbf{u}$  noté  $\tau_{\mathbf{u}}$ ,
- rotation de centre  $O$  d'angle orienté  $\theta$  noté  $\rho_{O,\theta}$ ,
- réflexion suivant un hyperplan  $\Delta$  noté  $\rho_{\Delta}$ .

Étant donné un pavage  $\mathbf{t}$  de  $\mathbb{R}^d$ , on note  $\mathcal{I}_{\mathbf{t}}$  le *groupe des isométries du pavage  $\mathbf{t}$* . Un pavage  $\mathbf{t}$  de  $\mathbb{R}^d$  est *périodique* s'il existe  $d$  vecteurs linéairement indépendants  $\mathbf{u}_1, \dots, \mathbf{u}_d$  tels que  $\tau_{\mathbf{u}_1}, \dots, \tau_{\mathbf{u}_d} \in \mathcal{I}_{\mathbf{t}}$ .

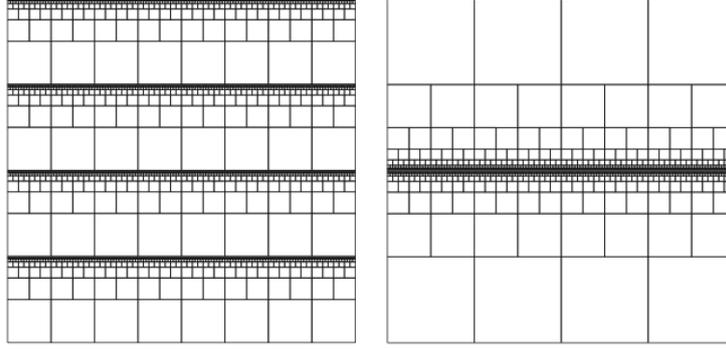


FIGURE 1.1 – La figure de gauche est un pavage alors que la figure de droite ne l'est pas car aucune tuile ne touche la ligne horizontale centrale.

**Théorème 1.2.1.** *Le groupe des isométries d'un pavage est discret.*

*Remarque.* Si la définition de tuile est modifiée, le groupe de symétrie n'est pas forcément discret.

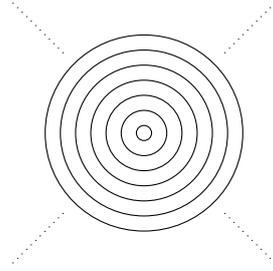


FIGURE 1.2 – Pavage avec tuiles non simplement connexes dont le groupe de symétrie n'est pas discret.

Soit  $\mathbf{t}$  un pavage périodique et soient  $\mathbf{u}_1, \dots, \mathbf{u}_d$   $d$  vecteurs linéairement indépendant tels que  $\tau_{\mathbf{u}_1}, \dots, \tau_{\mathbf{u}_d} \in \mathcal{I}_{\mathbf{t}}$ . On peut supposer qu'il n'existe pas de vecteur non nul  $\mathbf{u}$  colinéaire à  $\mathbf{u}_i$  pour un  $i \in [1, d]$  tel que la longueur de  $\mathbf{u}$  soit plus petite que celle de  $\mathbf{u}_i$ . Les vecteurs  $\mathbf{u}_1, \dots, \mathbf{u}_d$  engendrent un parallélépipède appelé *domaine fondamental* et  $\tau_{\mathbf{u}_1}, \dots, \tau_{\mathbf{u}_d}$  engendrent un sous-groupe  $\Gamma$  de  $\mathcal{I}_{\mathbf{t}}$  appelé réseau. Dans un pavage périodique le motif essentiel se trouve dans le domaine fondamental du réseau. Celui-ci peut aussi posséder une "certaine symétrie interne". Le groupe des isométries du pavage sera donc une "extension" du réseau  $\Gamma$  et d'un groupe fini. Ce dernier est constitué de réflexions ou de rotations; son action sur  $\Gamma$  limite le nombre des groupes paveurs.

### 1.2.2 Sous-groupe discret de $\mathcal{I}_2$

**Théorème 1.2.2** (Leonardo da Vinci's Theorem). *Un sous groupe fini de  $\mathcal{I}_2$  est soit le groupe cyclique  $C_n = \mathbb{Z}/n\mathbb{Z}$ , soit le groupe diédral  $D_n$*

*Démonstration.* Soit  $G$  un sous-groupe de  $\mathcal{I}_2$  fini.  $G$  ne contient ni translation ni symétrie glissée (la composée d'une translation et d'une réflexion d'axe parallèle au vecteur de translation) car ces éléments sont d'ordre infini et cela contredirait le fait que  $G$  est fini.

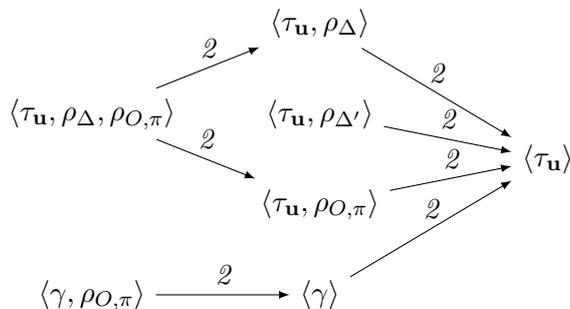
On suppose que  $G$  préserve l'orientation. Il ne contient donc que des rotations. Si deux rotations  $\rho_1$  et  $\rho_2$  avaient des centres différents, le commutateur  $\rho_2^{-1}\rho_1^{-1}\rho_2\rho_1$  serait une translation non triviale, ce qui n'est pas possible. Donc tous les éléments de  $G$  sont des rotations de même centre et par suite  $G$  est isomorphe à  $\mathbb{Z}/n\mathbb{Z}$  (ici  $n$  est l'ordre de  $G$ ).

Supposons que  $G$  contient au moins une réflexion. Les éléments qui préservent l'orientation forment un sous-groupe  $H$  non trivial isomorphe à  $C_n$  d'après ce qui précède. C'est le noyau du morphisme signature  $\epsilon : G \rightarrow \{-1, 1\}$  qui est non trivial et donc surjectif; il est donc d'indice 2 dans  $G$  et par suite normal;  $G$  s'identifie à un produit semi-direct  $C_n \rtimes \mathbb{Z}/2\mathbb{Z}$  qui est isomorphe au groupe diédral  $D_n$ .  $\square$

*Remarque.* Leonardo da Vinci a été le premier à avoir remarqué les symétries possibles d'un bâtiment central et comment y attacher des chapelles et des niches sans détruire les symétries du noyau. C'est la raison pour laquelle ce théorème lui revient.

Soit  $T$  le sous-groupe de  $\mathcal{I}_2$  engendré par les translations. Soit  $G$  un sous groupe discret de  $\mathcal{I}_2$ . On dit que  $G$  est un *groupe de frise* si  $G \cap T$  est isomorphe à  $\mathbb{Z}$  et que c'est un *groupe cristallographique* si  $G \cap T$  est isomorphe à  $\mathbb{Z}^2$ .

**Théorème 1.2.3.** *Soit  $G$  un groupe de frise dont les translations sont engendrées par  $\tau_{\mathbf{u}}$ . Il existe une droite  $\Delta$  de vecteur directeur  $\mathbf{u}$ , un point  $O \in \Delta$  et une droite  $\Delta'$  orthogonale à  $\Delta$  tels que  $G$  soit isomorphe à un des 7 groupes suivant représenté sous forme d'un treillis ou  $\xleftarrow{k}$  signifie "sous-groupe d'indice  $k$ " :*



où  $\gamma = \rho_{\Delta} \circ \tau_{\frac{\mathbf{u}}{2}}$  est une réflexion glissée (on remarque que  $\gamma^2 = \tau_{\mathbf{u}}$ ).

**Lemme 1.2.4.** *Un groupe cristallographique ne peut contenir des rotations que d'ordre 2, 3, 4 et 6 (c'est à dire d'angle  $\pi$ ,  $\frac{2\pi}{2}$ ,  $\frac{\pi}{4}$  et  $\frac{\pi}{3}$ ). De plus une rotation d'ordre 4 ne peut pas coexister avec une rotation d'ordre 3 ou 6.*

**Théorème 1.2.5.** *Il existe 17 groupes cristallographiques représenté par le treillis suivant :*



**Théorème 1.2.6.** *Un sous-groupe discret de  $\mathcal{I}_2$  est isomorphe à un groupe suivant :*

1. *un groupe cyclique  $C_n$  ou diédral  $D_n$ ,*
2. *un groupe de frise,*
3. *un sous groupe cristallographique.*

*Remarque.* Il existe 17 types de pavages périodiques qui se retrouvent presque tous dans l’alambrade de Grenade.

## 1.3 Ensemble de pavages

### 1.3.1 Relations d’équivalence entre pavages

Soit  $\mathbf{t} = (t_i)_{i \in \mathbb{N}}$  un pavage et  $h : \mathbb{R}^d \rightarrow \mathbb{R}^d$  un homéomorphisme. La famille de tuile  $h(\mathbf{t}) = (h(t_i))_{i \in \mathbb{N}}$  est un pavage, on dit que

- $\mathbf{t}$  et  $h(\mathbf{t})$  sont *topologiquement équivalents*,
- $\mathbf{t}$  et  $h(\mathbf{t})$  sont *congruents* si de plus  $h$  est une isométrie,
- $\mathbf{t}$  et  $h(\mathbf{t})$  sont *similaires* si de plus  $h$  est la composition d’une isométrie et d’une homothétie.

Les relations de topologiquement équivalent, congruence et similitude sont des relations d’équivalence.

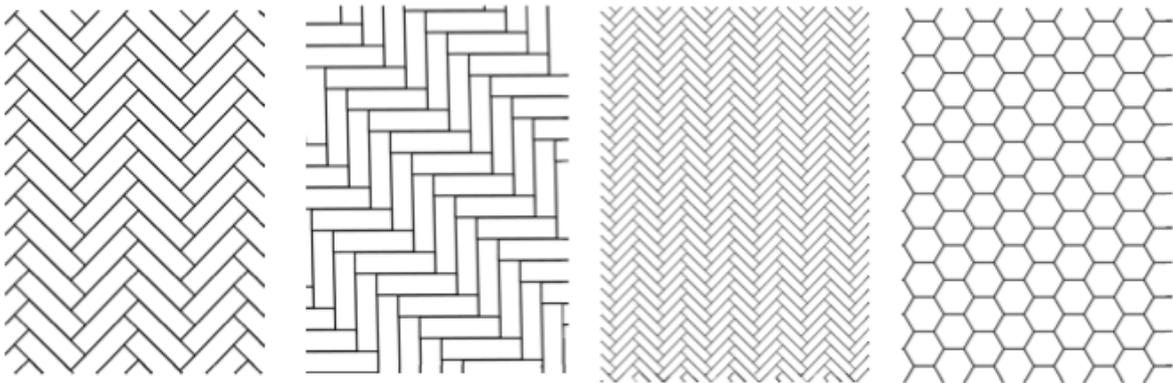


FIGURE 1.3 – Les deux premiers pavages sont congruents, les trois premiers sont similaires et les quatre pavages sont topologiquement équivalents.

### 1.3.2 Pavage par un nombre fini de tuiles “différentes”

On s’intéresse généralement aux pavages qui ont un nombre fini de tuiles. Deux tuiles  $t_1, t_2$  d’un pavage  $\mathbf{t}$  sont *équivalentes par translation* (respectivement *équivalentes par isométrie*) s’il existe une translation (respectivement une isométrie) qui transforme la première en la seconde. Si un pavage contient  $k$  classes d’équivalence on dit qu’il est *k-hedral par translation* (respectivement *k-hedral par isométrie*), si  $k = 1$  le pavage est dit monohedral. À chaque classe d’équivalence on associe un représentant appelé *proto-tuile par translation* (respectivement *proto-tuile par isométrie*). **Si on ne précise pas on se limitera à la classe d’équivalence par translation.**

Soit  $\mathcal{P} = \{p_1, \dots, p_k\}$  un ensemble de proto-tuiles de  $\mathbb{R}^d$ , on définit le *full-shift associé à  $\mathcal{P}$* , noté  $\mathbf{T}_{\mathcal{P}}$ , l’ensemble des pavages qui admet  $\mathcal{P}$  pour proto-tuiles. Un *motif  $m$*  d’un pavage  $\mathbf{t}$  est un ensemble fini de tuiles qui apparaît dans  $\mathbf{t}$  modulo une translation, on le note  $m \sqsubset \mathbf{t}$ . On note  $\mathcal{P}^*$  l’ensemble des motifs qui apparaissent dans tous les pavages de  $\mathbf{T}_{\mathcal{P}}$ .

## 1.4 Pavages par polygones réguliers

On considère ici des pavages de  $\mathbb{R}^2$ . Un pavage par polygones est dit *collé arêtes à arêtes* si l'intersection de deux tuiles est soit vide, soit un sommet soit une arête entière. Deux tuiles sont *voisines* si elles ont une arête en commun. Dans cette section on considère des pavages par polygones réguliers collés arête à arête.

Soit  $P$  un sommet adjacent à  $r$  polygones qui ont  $n_1, \dots, n_r$  arêtes données dans l'ordre inverse des aiguilles d'une montre. On dit que le sommet est de type  $n_1 \cdot n_2 \dots n_{r-1} \cdot n_r$ .

**Théorème 1.4.1.** *Un pavage monohedral par polygones réguliers collés arêtes à arêtes est régulier. C'est à dire s'il est constitué de triangles équilatéraux ou de carrés ou d'hexagone.*

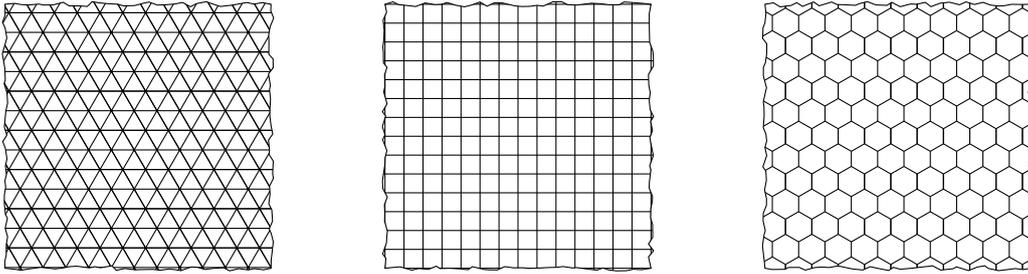


FIGURE 1.4 – Pavages réguliers.

*Démonstration.* L'angle intérieur d'un sommet d'un  $n$ -gone est  $\pi \left(1 - \frac{2}{n}\right)$ . Ainsi pour un sommet à  $r$  voisins de type  $n_1 \cdot n_2 \dots n_{r-1} \cdot n_r$  on a

$$\sum_{i=1}^r \left(1 - \frac{2}{n_i}\right) = 2. \quad (1.1)$$

Si les tuiles du pavage sont des  $n$ -gones réguliers, on en déduit que  $n = \frac{2r}{r-2}$ . Comme  $n \geq 3$ , on en déduit que  $r \leq 6$  et comme  $n$  est entier, on a  $n = 3, 4$  ou  $6$  d'où le résultat.  $\square$

Si on résout l'équation (1.1) il existe 21 types de sommets différents pour un pavage par polygones. Un pavage par polygones réguliers collés arêtes à arêtes est *archimédien* si tous les sommets sont de même type.

**Théorème 1.4.2** (Kepler 1619). *Il existe exactement onze pavages par polygones réguliers collés arêtes à arêtes archimédiens.*

## 1.5 Règles locales

### 1.5.1 Ensemble de pavage de type fini

Soit  $\mathcal{P}$  un ensemble de proto-tuile. Soit  $\mathcal{F} \subset \mathcal{P}^*$  on définit l'ensemble de pavage de motifs interdits  $\mathcal{F}$  :

$$\mathbf{T}_{\mathcal{F}} = \{\mathbf{t} \in \mathbf{X}_{\mathcal{P}} : m \text{ n'apparaît pas dans } \mathbf{t} \text{ pour tout } m \in \mathcal{F}\}$$

Un ensemble de pavage  $\mathbf{T}$  est de *type fini* s'il existe  $\mathcal{F} \subset \mathcal{P}^*$  fini tel que  $\mathbf{T} = \mathbf{T}_{\mathcal{F}}$ .

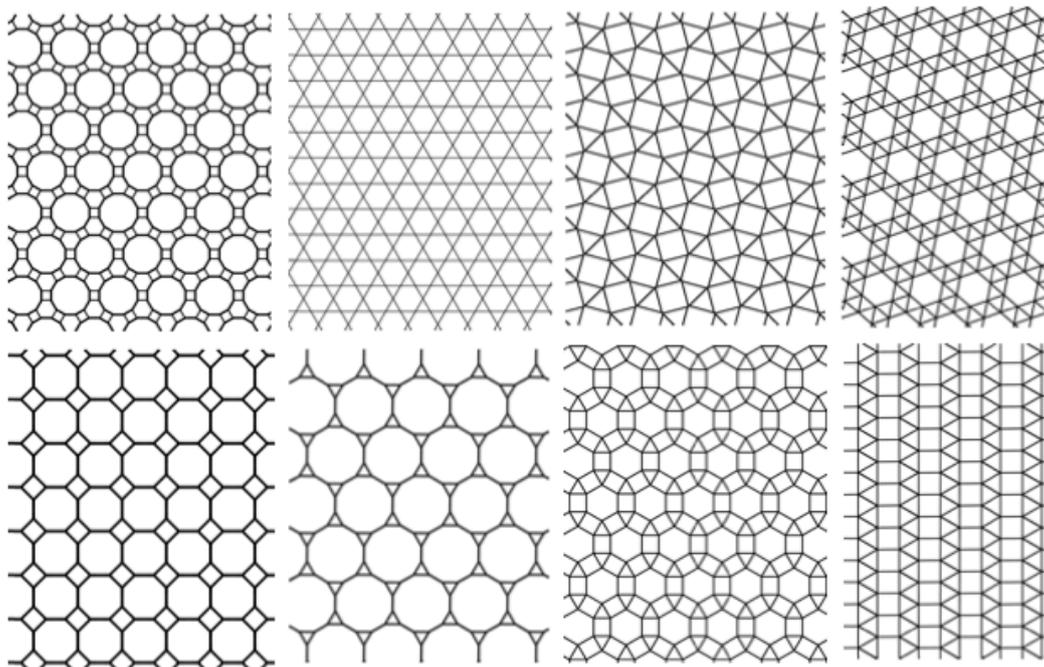


FIGURE 1.5 – Quelques pavages archimédiens

### 1.5.2 Tuiles de Wang

Wang [Wan61] considère les pavages formés de tuiles carrés dont les côtés sont coloriés et deux tuiles peuvent être assemblées suivant un côté si et seulement si les deux côtés ont la même couleur. Il pose la question de savoir s'il existe un jeu de tuile qui pave aperiodiquement.

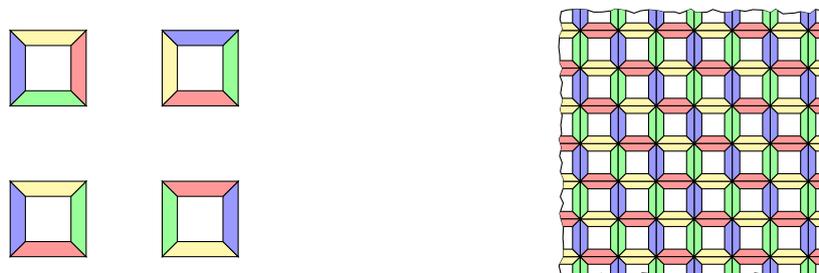


FIGURE 1.6 – Un jeu de tuiles de Wang et un pavage associé. On remarque que tout pavage associé à ce jeu est un pavage périodique utilisant uniquement les deux tuiles du bas.

### 1.5.3 Tuiles décorées

Soit  $\mathcal{P}$  un ensemble de proto-tuiles et  $\mathcal{B}$  un alphabet fini. Un *pavage colorié* est un pavage  $\mathbf{t} = (t_i)_{\mathbb{N}}$  auquel à chaque tuile on associe une couleur de  $\mathcal{B}$ , on le note  $\mathbf{t} = (t_i, b_i)_{\mathbb{N}}$ . On note  $\mathbf{X}'_{\mathcal{P} \times \mathcal{B}}$  l'ensemble de tous les pavages coloriés et  $\pi : \mathbf{X}'_{\mathcal{P} \times \mathcal{B}} \rightarrow \mathbf{X}_{\mathcal{P}}$  la projection qui supprime la couleur ajoutée.

Un ensemble de pavage  $\mathbf{T} \subset \mathbf{X}_{\mathcal{P}}$  est *sofique* s'il existe un ensemble fini  $\mathcal{F}'$  de motifs colorés

tel que l'ensemble de pavage de type fini

$$\mathbf{T}'_{\mathcal{F}'} = \{\mathbf{t} \in \mathbf{X}'_{\mathcal{P}} : m \text{ n'apparaît pas dans } \mathbf{t} \text{ pour tout } m \in \mathcal{F}'\}$$

vérifie  $\pi(\mathbf{T}'_{\mathcal{F}'}) = \mathbf{T}_{\mathcal{F}}$ .

*Remarque.* Quitte à modifier la forme d'une tuile, on peut toujours ramener une tuile décorée à une tuile géométrique.



FIGURE 1.7 – Modification d'un jeu de tuile de Wang pour n'avoir que des tuiles géométriques.

## Chapitre 2

# Éléments de dynamique symbolique

Dans ce chapitre on se limite aux pavages par  $d$ -cubes collés arêtes à arêtes. Au lieu de s'intéresser à un pavage en particulier, on va considérer un ensemble de pavages sur lequel on met une dynamique. Les notions vues dans ce chapitre se généralisent à des espaces de pavages dont les tuiles ne sont pas nécessairement des  $d$ -cubes.

### 2.1 Sous-shift

#### 2.1.1 Motifs d'une configuration

Soit  $\mathcal{A}$  un alphabet fini. On considère l'ensemble des fonctions  $x : \mathbb{Z}^d \rightarrow \mathcal{A}$ , noté  $\mathcal{A}^{\mathbb{Z}^d}$ . Un élément  $x \in \mathcal{A}^{\mathbb{Z}^d}$  est une *configuration*. On peut le voir comme un pavage par cubes de dimension  $d$ .

Soit  $\mathbb{U} \subset \mathbb{Z}^d$  un ensemble fini. Un *motif* est un élément  $p \in \mathcal{A}^{\mathbb{U}}$ , le *support* de  $p$  est  $\text{supp}(p) = \mathbb{U}$ . Étant donnée une configuration  $x \in \mathcal{A}^{\mathbb{Z}^d}$ , on note  $x_{\mathbb{U}}$  le motif correspondant à la restriction de  $x$  à  $\mathbb{U}$ . Un motif  $p \in \mathcal{A}^{\mathbb{U}}$  *apparaît* dans la configuration  $x$  s'il existe  $i \in \mathbb{Z}^d$  tel que  $x_{i+\mathbb{U}} = p$ , on le note  $p \sqsubset x$ .

Pour  $\mathbb{U} \subset \mathbb{Z}^d$  fini, le *langage de support*  $\mathbb{U}$  de l'ensemble  $\mathbf{T} \subset \mathcal{A}^{\mathbb{Z}^d}$  est

$$\mathcal{L}_{\mathbb{U}}(\mathbf{T}) = \{p \in \mathcal{A}^{\mathbb{U}^n} : \text{il existe } x \in \mathbf{T} \text{ tel que } p \sqsubset x\}$$

Pour  $n \in \mathbb{N}$ , on note  $\mathbb{U}_n = [0, n-1]^d$  le *support élémentaire*. Un *langage (élémentaire)  $\mathcal{L}$*  est un sous-ensemble de motifs de support élémentaire, c'est à dire  $\mathcal{L} \subset \cup_{n \in \mathbb{N}} \mathcal{A}^{\mathbb{U}_n}$ . Étant donné un ensemble  $\mathbf{T} \subset \mathcal{A}^{\mathbb{Z}^d}$ , on note  $\mathcal{L}_n(\mathbf{T}) = \{p \in \mathcal{A}^{\mathbb{U}_n} : \text{il existe } x \in \mathbf{T} \text{ tel que } p \sqsubset x\}$  le *langage d'ordre  $n$*  de  $\mathbf{T}$  et  $\mathcal{L}(\mathbf{T}) = \cup_{n \in \mathbb{N}} \mathcal{L}_n(\mathbf{T})$  le *langage* de  $\mathbf{T}$ .

*Exemple 2.1.1.* Si on considère la configuration suivante

$$x = \begin{array}{|c|c|c|c|c|c|} \hline 1 & 0 & 1 & 0 & 1 & 0 \\ \hline 0 & 1 & 0 & 1 & 0 & 1 \\ \hline 1 & 0 & 1 & 0 & 1 & 0 \\ \hline 0 & 1 & 0 & 1 & 0 & 1 \\ \hline 1 & 0 & 1 & 0 & 1 & 0 \\ \hline 0 & 1 & 0 & 1 & 0 & 1 \\ \hline \end{array}$$

$$\text{on a : } \mathcal{L}_1(\{x\}) = \left\{ \begin{array}{|c|} \hline 0 \\ \hline \end{array}, \begin{array}{|c|} \hline 1 \\ \hline \end{array} \right\}, \mathcal{L}_2(\{x\}) = \left\{ \begin{array}{|c|c|} \hline 1 & 0 \\ \hline 0 & 1 \\ \hline \end{array}, \begin{array}{|c|c|} \hline 0 & 1 \\ \hline 1 & 0 \\ \hline \end{array} \right\}, \mathcal{L}_3(\{x\}) = \left\{ \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 1 & 0 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}, \begin{array}{|c|c|c|} \hline 1 & 0 & 1 \\ \hline 0 & 1 & 0 \\ \hline 1 & 0 & 1 \\ \hline \end{array} \right\}$$

On peut aussi s'intéresser au *langage rectangulaire* :

$$\mathcal{L}^{\text{rect}}(\mathbf{T}) = \bigcup_{(n_1, \dots, n_d) \in \mathbb{N}^d} \mathcal{L}_{(n_1, \dots, n_d)}^{\text{rect}}(\mathbf{T})$$

où  $\mathcal{L}_{(n_1, \dots, n_d)}^{\text{rect}}(\mathbf{T}) = \{p \in \mathcal{A}^{[0, n_1-1] \times \dots \times [0, n_d-1]} : \text{il existe } x \in \mathbf{T} \text{ tel que } p \sqsubset x\}$ ,

ou encore au *langage généralisé* :

$$\mathcal{L}^{\text{gen}}(\mathbf{T}) = \bigcup_{\substack{\mathbb{U} \subset \mathbb{Z}^d \\ \text{fini}}} \mathcal{L}_{\mathbb{U}}(\mathbf{T}) = \bigcup_{\substack{\mathbb{U} \subset \mathbb{Z}^d \\ \text{fini}}} \{p \in \mathcal{A}^{\mathbb{U}} : \text{il existe } x \in \mathbf{T} \text{ tel que } p \sqsubset x\}.$$

Étant donné un motif  $p \in \mathcal{A}^{\mathbb{U}}$ , on définit le *cylindre* centré sur  $p$  par :

$$[p] = [p]_{\mathbb{U}} = \{x \in \mathcal{A}^{\mathbb{Z}^d} : x_{\mathbb{U}} = p\}.$$

Pour  $k \in \mathbb{Z}^d$ , on note  $[p]_k = \{x \in \mathcal{A}^{\mathbb{Z}^d} : x_{k+\mathbb{U}} = p\}$ .

### 2.1.2 Topologie de Cantor

L'alphabet  $\mathcal{A}$  étant fini, on peut le munir de la topologie discrète. L'ensemble  $\mathcal{A}^{\mathbb{Z}^d}$  est naturellement muni par la topologie produit. Une base d'ouvert pour cette topologie est formée de l'ensemble des cylindres  $\{[p]_{\mathbb{U}} : \mathbb{U} \subset \mathbb{Z}^d \text{ fini et } p \in \mathcal{A}^{\mathbb{U}}\}$ . Si  $(\mathbb{U}_n)_{n \in \mathbb{N}}$  est une suite croissante de sous-ensembles finis de  $\mathbb{Z}^d$  telle que  $\bigcup_n \mathbb{U}_n = \mathbb{Z}^d$ , on peut aussi prendre comme base  $\{[p]_{\mathbb{U}_n} : n \in \mathbb{N} \text{ et } p \in \mathcal{A}^{\mathbb{U}_n}\}$ .

Pour cette topologie, un cylindre est un ouvert-fermé car pour un motif  $p \in \mathcal{A}^{\mathbb{U}}$ , on a :

$$\mathcal{A}^{\mathbb{Z}^d} \setminus [p]_{\mathbb{U}} = \bigcup_{q \in \mathcal{A}^{\mathbb{U}} \setminus \{p\}} [q]_{\mathbb{U}}$$

L'espace  $\mathcal{A}^{\mathbb{Z}^d}$  est donc totalement discontinu, c'est à dire que les seuls ensembles connexes sont l'ensemble vide et les singletons.

Dans cette topologie, si on considère une suite  $(c^i)_{i \in \mathbb{N}}$  d'éléments de  $\mathcal{A}^{\mathbb{Z}^d}$ , on dit que la suite converge vers  $c \in \mathcal{A}^{\mathbb{Z}^d}$  si pour tout  $n \in \mathbb{Z}^d$  il existe  $k \in \mathbb{N}$  tel que pour tout  $i \geq k$  on ait  $c_n^i = c_n$ . En d'autres termes étant donné une position, à partir d'un certain moment, une seule tuile apparaîtra dans cette position.

**Proposition 2.1.1.**  *$\mathcal{A}^{\mathbb{Z}^d}$  est compact pour la topologie produit.*

L'espace  $\mathcal{A}$  étant fini, il est compact, donc par le théorème de Tychonov, l'espace produit  $\mathcal{A}^{\mathbb{Z}^d}$  est compact. Cela peut aussi se démontrer simplement en utilisant la définition de la topologie :

*Démonstration.* Soient  $(c^i)_{i \in \mathbb{N}}$  une suite de configuration de  $\mathcal{A}^{\mathbb{Z}^d}$  et  $(n_i)_{i \in \mathbb{Z}^d}$  une énumération de  $\mathbb{Z}^d$ .

On choisit les indices  $i_0 < i_1 < i_2 < i_3 < \dots$  inductivement de la manière suivante :

- $i_0$  est le plus petit entier tel qu'il y ait une infinité d'indice  $i \in \mathbb{N}$  tel que  $c_{n_0}^{i_0} = c_{n_0}^i$  ;
- pour tout  $j \geq 1$ ,  $i_j > i_{j-1}$  est le plus petit entier tel que  $c_{n_k}^{i_j} = c_{n_k}^{i_{j-1}}$  pour tout  $k \leq j-1$  et tel qu'il y ait une infinité d'indice  $i \in \mathbb{N}$  tel que  $c_{n_k}^{i_j} = c_{n_k}^i$  pour tout  $k \leq j$ .

La suite extraite  $(c_{n_k}^{i_j})_{j \in \mathbb{N}}$  converge.

□

De plus cet espace est métrisable en choisissant la distance suivante

$$d(x, y) = 2^{-\min\{|i| : x_i \neq y_i\}} \text{ pour tout } x, y \in \mathcal{A}^{\mathbb{Z}^d} \text{ où } |i| \text{ est la plus grande coordonnée de } i.$$

Cela signifie que deux configurations sont proches si elles sont semblables sur un grand domaine autour de l'origine.

### 2.1.3 Sous-shifts

Le groupe  $\mathbb{Z}^d$  agit naturellement par décalage défini pour tout  $j \in \mathbb{Z}^d$  par :

$$\begin{aligned} \sigma_{\mathcal{A}^{\mathbb{Z}^d}}^j : \quad \mathcal{A}^{\mathbb{Z}^d} &\longrightarrow \mathcal{A}^{\mathbb{Z}^d} \\ x = (x_i)_{i \in \mathbb{Z}^d} &\longmapsto \sigma^j(x) = (x_{i+j})_{i \in \mathbb{Z}^d}. \end{aligned}$$

Un *sous-shift* est un sous ensemble fermé de  $\mathcal{A}^{\mathbb{Z}^d}$  stable par décalage.

*Exemple 2.1.2.* Soit  $x \in \{0, 1\}^{\mathbb{Z}}$  tel que  $x_j = 0$  si  $j \neq 0$  et  $x_0 = 1$ . L'ensemble  $\mathcal{O}(x) = \{\sigma^n(x) : n \in \mathbb{Z}\}$  est shift invariant. Par contre la suite  $(\sigma^i(x))_{i \in \mathbb{N}}$  converge vers la configuration  ${}^\infty 0^\infty$  donc  $\mathcal{O}(x) = \{\sigma^n(x) : n \in \mathbb{Z}\}$  n'est pas fermé. Par contre  $\overline{\mathcal{O}(x)} = \{\sigma^n(x) : n \in \mathbb{Z}\} \cup \{{}^\infty 0^\infty\}$  est un sous-shift.

### 2.1.4 Facteur et conjugaison

Soient  $\mathbf{T} \subset \mathcal{A}^{\mathbb{Z}^d}$  et  $\mathbf{T}' \subset \mathcal{B}^{\mathbb{Z}^d}$ . Une fonction  $F : \mathbf{T} \rightarrow \mathbf{T}'$  est un *morphisme* si elle est continue et commute avec le shift, c'est à dire  $F \circ \sigma_{\mathbf{T}}^i = \sigma_{\mathbf{T}'}^i \circ F$  pour tout  $i \in \mathbb{Z}^d$ .

Une fonction  $F : \mathbf{T} \rightarrow \mathbf{T}'$  est *définie localement* ou *définie par blocs* s'il existe  $\mathbb{U} \subset \mathbb{Z}^d$  sous-ensemble fini, appelé *voisinage*, et une fonction  $\overline{F} : \mathcal{L}_{\mathbb{U}}(\mathbf{T}) \rightarrow \mathcal{L}_{\{0\}}(\mathbf{T}')$ , appelée *fonction locale* telle que  $F(x)_i = \overline{F}((x_{i+j})_{j \in \mathbb{U}})$  pour tout  $i \in \mathbb{Z}^d$  et  $x \in \mathbf{T}$ . Étant donnée une fonction locale  $\overline{F} : \mathbf{T} \rightarrow \mathbf{T}'$ , on peut l'étendre à l'ensemble fini  $\mathbb{U}' \subset \mathbb{Z}^d$  comme une fonction  $\overline{F} : \mathcal{L}_{\mathbb{U}+\mathbb{U}'}(\mathbf{T}) \rightarrow \mathcal{L}_{\mathbb{U}'}(\mathbf{T}')$  définie par  $\overline{F}(x)_i = \overline{F}((x_{i+j})_{j \in \mathbb{U}'})$  pour tout  $i \in \mathbb{U}'$  et  $x \in \mathbf{T}$ .

**Théorème 2.1.2** ([Hed69]). *Soient  $\mathbf{T} \subset \mathcal{A}^{\mathbb{Z}^d}$  et  $\mathbf{T}' \subset \mathcal{B}^{\mathbb{Z}^d}$ . Une fonction  $F : \mathbf{T} \rightarrow \mathbf{T}'$  est continue et invariante par décalage ( $\sigma^j \circ F = F \circ \sigma^j$  pour tout  $j \in \mathbb{Z}^d$ ) si et seulement si elle est définie localement.*

*Démonstration.* Soit  $F : \mathbf{T} \rightarrow \mathbf{T}'$  continue et qui commute avec le décalage. Comme  $F$  est continue sur un compact, elle est uniformément continue par le théorème de Heine. Il existe donc  $r \in \mathbb{N}$  tel que si  $d(x, y) \leq 2^{-r}$  alors  $F(x)_0 = F(y)_0$  pour  $x, y \in \mathbf{T}$ . Soit  $\mathbb{U} = \{i \in \mathbb{Z}^d : |i| \leq r\}$ . On peut alors définir :

$$\begin{aligned} \overline{F} : \mathcal{L}_{\mathbb{U}}(\mathbf{T}) &\longrightarrow \mathcal{L}_{\{0\}}(\mathbf{T}') \\ u &\longmapsto F(x)_0 \text{ où } x \in [u]_{\mathbb{U}} \end{aligned}$$

Par  $\sigma$  invariance, la fonction définie par blocs par  $\mathbb{U}$  et  $\overline{F}$  correspond à  $F$ .

Soit  $F : \mathbf{T} \rightarrow \mathbf{T}'$  une fonction définie par blocs par  $\mathbb{U}$  et  $\overline{F}$ . Soit  $r = \max\{|i| : i \in \mathbb{U}\}$ . Pour tous  $x, y \in \mathbf{T}$  tels que  $d(x, y) = 2^{-n-r}$ , on a  $x_{\mathbb{U}_n+\mathbb{U}} = y_{\mathbb{U}_n+\mathbb{U}}$  où  $\mathbb{U}_n = \{i \in \mathbb{Z}^d : |i| \leq n\}$  d'où  $F(x)_{\mathbb{U}_n} = F(y)_{\mathbb{U}_n}$  c'est à dire  $d(F(x), F(y)) \leq 2^r d(x, y)$ . Ainsi  $F$  est lipschitzienne et a fortiori continue. De plus, pour tout  $x \in \mathbf{T}$  et tout  $i \in \mathbb{Z}^d$ , on a  $F(\sigma(x))_i = \overline{F}((\sigma(x)_{i+u})_{u \in \mathbb{U}}) = \overline{F}((x_{1+i+u})_{u \in \mathbb{U}}) = \sigma(F(x))_i$ . La fonction  $F$  commute donc avec  $\sigma$ .  $\square$

*Exemple 2.1.3.* Si  $\mathcal{A} = \mathbb{Z}/2\mathbb{Z}$ ,  $\mathcal{A}^{\mathbb{Z}^d}$  hérite de la structure de groupe par passage au produit, la fonction  $F(x) = x + \sigma(x)$  est continue et shift-invariante, elle peut être définie localement par le voisinage  $\mathbb{U} = \{0, 1\}$  et la fonction locale  $\overline{F}(x_0, x_1) = x_0 + x_1$ .

**Corollaire 2.1.3.** *L'image d'un décalage par une fonction de bloc est également un décalage.*

**Corollaire 2.1.4.** *Si  $F : \mathbf{T} \rightarrow \mathbf{T}'$  est définie localement et bijective, alors  $F^{-1}$  est définie localement.*

*Démonstration.* Si  $F$  est bijective et continue sur un compact alors  $F^{-1}$  est continue.  $\square$

Soient  $\mathbf{T} \subset \mathcal{A}^{\mathbb{Z}^d}$ ,  $\mathbf{T}' \subset \mathcal{B}^{\mathbb{Z}^d}$  et  $\varphi : \mathcal{A}^{\mathbb{Z}^d} \rightarrow \mathcal{B}^{\mathbb{Z}^d}$  une fonction bloc (ou morphisme). Si  $\varphi : \mathbf{T} \rightarrow \mathbf{T}'$  est bijectif alors  $\varphi$  est une *conjugaison* et les sous-shifts  $\mathbf{T}$  et  $\mathbf{T}'$  sont *conjugués*. Si  $\varphi : \mathbf{T} \rightarrow \mathbf{T}'$  est surjectif alors  $\varphi$  est un *facteur* et  $\mathbf{T}'$  est un *facteur* de  $\mathbf{T}$ .

### 2.1.5 Liens avec les systèmes dynamiques

Soient  $X$  un espace métrique compact et  $F$  une  $\mathbb{Z}^d$  action continue sur  $X$ . On dit que  $(X, F)$  est un *système dynamique discret*.

Soit  $\mathcal{V}$  une partition de  $X$ . On définit

$$\begin{aligned} \varphi : X &\longrightarrow \mathcal{V}^{\mathbb{Z}^d} \\ x &\longmapsto y \quad \text{tel que } F^n(x) \in y_n \text{ pour tout } n \in \mathbb{Z}^d \end{aligned}$$

L'ensemble  $\varphi(X) = \mathbf{T}_{(X,F)} = \{y \in \mathcal{V}^{\mathbb{Z}^d} : \exists x \in X \text{ tel que } F^n(x) \in y_n\}$  est un sous-shift. Pour tout  $n \in \mathbb{Z}^d$ , on a  $\varphi(F^n(x)) = \sigma^n(\varphi(x))$ .

On dit que  $\mathcal{V}$  est *génératrice* si pour tout  $(y_i)_{i \in \mathbb{Z}^d}$  l'ensemble  $\bigcap_{n \in \mathbb{Z}^d} F^{-n}(y_n)$  est vide ou réduit à un singleton. On vérifie les propriétés suivantes :

- si  $\mathcal{V}$  est génératrice alors  $\varphi$  est une bijection ;
- si  $\mathcal{V}$  est formée par des ouvert-fermés alors  $\varphi$  est continue ;
- si  $\mathcal{V}$  est génératrice et formée par des ouvert-fermés alors  $\varphi$  est une conjugaison.

Un système dynamique  $(X, F)$  est *expansif* s'il existe  $\epsilon > 0$  tel que pour tous  $x, y \in X$ ,  $x \neq y$  alors il existe  $n \in \mathbb{Z}^d$  tel que  $d(F^n(x), F^n(y)) \geq \epsilon$ . Il est facile de voir qu'un système dynamique expansif admet une partition génératrice. On en déduit que si  $(X, F)$  est un système dynamique expansif tel que  $X$  est totalement discontinue alors  $(X, F)$  est conjugué à un sous-shift.

## 2.2 Sous-shift de type fini et sofiques

### 2.2.1 Sous-shift défini par motifs interdits

Soit  $\mathcal{F}$  un ensemble de motifs. On définit le *sous-shift des motifs interdits*  $\mathcal{F}$  sur l'alphabet  $\mathcal{A}$  par

$$\mathbf{T}(\mathcal{A}, d, \mathcal{F}) = \left\{ x \in \mathcal{A}^{\mathbb{Z}^d} : p \text{ n'apparaît pas dans } x \text{ pour tout } p \in \mathcal{F} \right\} = \mathcal{A}^{\mathbb{Z}^d} \setminus \bigcup_{p \in \mathcal{F}, i \in \mathbb{Z}^d} \sigma^{-i}([p]).$$

On vérifie que  $\mathbf{T}(\mathcal{A}, d, \mathcal{F})$  est bien un sous-shift.

**Proposition 2.2.1.** *Étant donné un ensemble  $\mathcal{F}$  de motifs si pour tout  $n$  il existe un motif de support  $[-n, n]^d$  qui ne contient pas de motif de  $\mathcal{F}$  alors  $\mathbf{T}(\mathcal{A}, d, \mathcal{F})$  est non vide.*

*Démonstration.* On utilise la compacité de  $\mathcal{A}^{\mathbb{Z}^d}$ . □

**Proposition 2.2.2.** *Un ensemble  $\mathbf{T} \subset \mathcal{A}^{\mathbb{Z}^d}$  est un sous-shift si et seulement si il existe un ensemble de motifs  $\mathcal{F}$  tel que  $\mathbf{T} = \mathbf{T}(\mathcal{A}, d, \mathcal{F})$ .*

*Démonstration.* Soit  $\mathbf{T}$  un sous-shift, on considère l'ensemble des motifs  $\mathcal{F}$  égal au complémentaire du langage de  $\mathbf{T}$ , c'est à dire  $\mathcal{F} = \bigcup_{n \in \mathbb{N}} \mathcal{A}^{\mathbb{U}_n} \setminus \mathcal{L}(\mathbf{T})$ . Si  $x \in \mathbf{T}$ , tout motif de  $x$  est un sous-motif de  $\mathcal{L}(\mathbf{T})$  donc  $x \in \mathbf{T}(\mathcal{A}, d, \mathcal{F})$ . Si  $x \in \mathbf{T}(\mathcal{A}, d, \mathcal{F})$  alors pour tout  $n \in \mathbb{N}$  on a  $x_{[-n, n]^d} \in \mathcal{L}(\mathbf{T})$  c'est à dire il existe  $y^n \in \mathbf{T}$  tel que  $x_{[-n, n]^d} = y^n_{[-n, n]^d}$ . On a donc  $\lim_{n \rightarrow \infty} y^n = x$  et comme  $\mathbf{T}$  est fermé,  $x \in \mathbf{T}$ . □

### 2.2.2 Sous-shift de type fini

L'ensemble  $\mathbf{T} \subset \mathcal{A}^{\mathbb{Z}^d}$  est un *sous-shift de type fini* s'il existe un ensemble fini de motifs  $\mathcal{F}$  tel que  $\mathbf{T} = \mathbf{T}(\mathcal{A}, d, \mathcal{F})$ . De manière équivalente on peut définir un sous-shift de type fini par des motifs autorisés.

Étant donné un sous-shift de type fini  $\mathbf{T}(\mathcal{A}, d, \mathcal{F})$ , on peut supposer qu'il existe  $n \in \mathbb{N}$  tel que  $\mathcal{F} \subset \mathcal{A}^{\mathbb{U}_n}$ . Le plus petit  $n$  qui vérifie cette propriété est appelée *l'ordre* de  $\mathbf{T}(\mathcal{A}, d, \mathcal{F})$ .

### Quelques exemples

*Exemple 2.2.1* (L'échiquier). Soient  $n \geq 2$  et  $\mathcal{A}_n = \{0, 1, \dots, n-1\}$ . On considère  $\mathcal{F}_n = \left\{ \begin{array}{|c|} \hline i \\ \hline i \\ \hline \end{array}, \begin{array}{|c|c|} \hline i & i \\ \hline \end{array} : i \in \mathcal{A}_n \right\}$  et on note  $\mathbf{T}_n = \mathbf{T}(\mathcal{A}_n, 2, \mathcal{F}_n) \subset \mathcal{A}_n^{\mathbb{Z}^2}$ .

Pour  $n = 2$ ,  $\mathbf{T}_2$  a deux configurations :

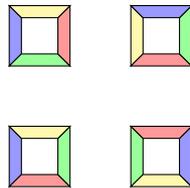
$$\mathbf{T}_2 = \left\{ \begin{array}{|c|c|c|c|c|c|} \hline 0 & 1 & 0 & 1 & 0 & 1 \\ \hline 1 & 0 & 1 & 0 & 1 & 0 \\ \hline 0 & 1 & 0 & 1 & 0 & 1 \\ \hline 1 & 0 & 1 & 0 & 1 & 0 \\ \hline 0 & 1 & 0 & 1 & 0 & 1 \\ \hline 1 & 0 & 1 & 0 & 1 & 0 \\ \hline \end{array}, \begin{array}{|c|c|c|c|c|c|} \hline 0 & 1 & 0 & 1 & 0 & 1 \\ \hline 1 & 0 & 1 & 0 & 1 & 0 \\ \hline 0 & 1 & 0 & 1 & 0 & 1 \\ \hline 1 & 0 & 1 & 0 & 1 & 0 \\ \hline 0 & 1 & 0 & 1 & 0 & 1 \\ \hline 1 & 0 & 1 & 0 & 1 & 0 \\ \hline \end{array} \right\}$$

Pour  $n = 3$ , il existe des configurations "gelées", c'est à dire qui ne sont pas dans  $\mathbf{T}_3$  si elles sont modifiées en un nombre fini de sites. Ce sont les extensions périodiques des points suivants :

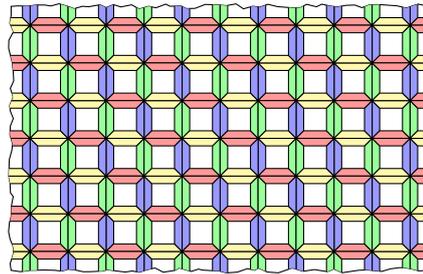
$$\begin{array}{|c|c|c|c|c|c|} \hline 2 & 0 & 1 & 2 & 0 & 1 \\ \hline 1 & 2 & 0 & 1 & 2 & 0 \\ \hline 0 & 1 & 2 & 0 & 1 & 2 \\ \hline 2 & 0 & 1 & 2 & 0 & 1 \\ \hline 1 & 2 & 0 & 1 & 2 & 0 \\ \hline 0 & 1 & 2 & 0 & 1 & 2 \\ \hline 2 & 0 & 1 & 2 & 0 & 1 \\ \hline \end{array}, \begin{array}{|c|c|c|c|c|c|} \hline 1 & 0 & 2 & 1 & 0 & 2 \\ \hline 2 & 1 & 0 & 2 & 1 & 0 \\ \hline 0 & 2 & 1 & 0 & 2 & 1 \\ \hline 1 & 0 & 2 & 1 & 0 & 2 \\ \hline 2 & 1 & 0 & 2 & 1 & 0 \\ \hline 0 & 2 & 1 & 0 & 2 & 1 \\ \hline 1 & 0 & 2 & 1 & 0 & 2 \\ \hline \end{array}$$

*Exemple 2.2.2* (Wang tiles). On considère un ensemble  $T$  fini de tuiles distinctes formées de carrés de taille  $1 \times 1$  dont les arêtes sont colorées et qu'aucune arête horizontale n'a la même couleur qu'une arête verticale (ou bien l'on suppose que l'on ne peut pas coller une arête horizontale sur une arête verticale). Pour une tuile  $t \in T$ , on note respectivement  $\bar{t}, |t, \underline{t}$  et  $t|$  respectivement les couleurs de l'arête du haut, de droite, du bas et de gauche de  $t$ . On note  $C(T) = \{\bar{t}, |t, \underline{t}, t| : t \in T\}$  l'ensemble des couleurs colorant les tuiles de  $T$ . Un pavage de Wang est un recouvrement de  $\mathbb{R}^2$  par des tuiles de  $T$  tel que les arêtes de deux tuiles adjacentes coïncident si et seulement si elles ont la même couleur.

Jeu de tuiles



Un pavage associé



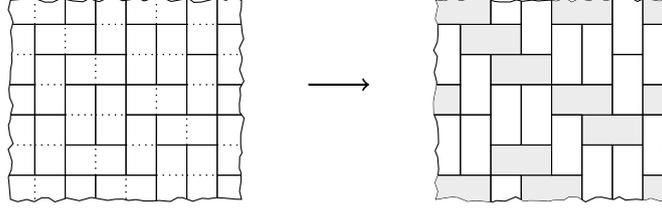
On identifie chaque pavage par des tuiles de Wang par une configuration  $w = (w_n)_{n \in \mathbb{Z}^2} \in T^{\mathbb{Z}^2}$ . On définit alors le sous-shift de Wang par :

$$\mathcal{W}_T = \left\{ (w_n)_{n \in \mathbb{Z}^2} \in T^{\mathbb{Z}^2} : w_{i,j} = |w_{i+1,j} \text{ et } \underline{w_{i,j}} = \bar{w_{i,j-1}} \right\}$$

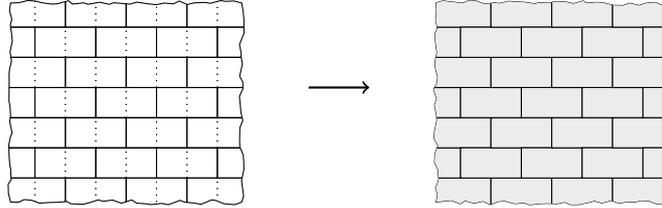
*Exemple 2.2.3* (Domino). On considère le jeu de tuiles de Wang suivant :

$$T_D = \left\{ \begin{array}{c} \square \\ \square \\ \square \\ \square \end{array} \right\}$$

La figure suivante considère une configuration de  $\mathcal{W}_{T_D}$  auquel on a associé un pavage par des dominos  $1 \times 2$  et  $2 \times 1$ .



Le sous-shift  $\mathcal{W}_{T_D}$  admet aussi des configurations “gelées” comme la suivante :



Exemple 2.2.4 (3-dots).

$$\mathbf{T} = \{x \in \{0, 1\}^{\mathbb{Z}^2} : \forall (i, j) \in \mathbb{Z}^2, x_{(i,j)} + x_{(i+1,j)} + x_{(i,j+1)} = 0 \pmod{2}\}$$

### Le cas de la dimension 1

Un graphe orienté  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  est la donnée d’un ensemble  $\mathcal{V}$  de sommets et d’un ensemble  $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$  d’arêtes orientées. Pour une arête  $e \in \mathcal{E}$ , on note  $i(e)$  le sommet initial et  $t(e)$  le sommet terminal de l’arête, en d’autres termes,  $e = (i(e), t(e)) \in \mathcal{E}$ . On définit le *sous-shift de type fini associé au graphe  $\mathcal{G}$*  par :

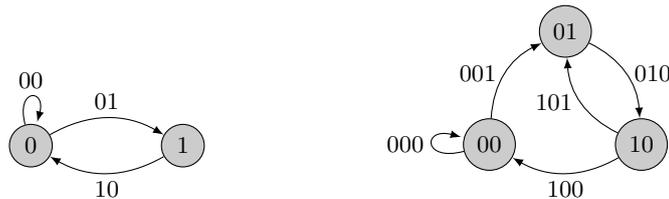
$$\mathbf{T}_{\mathcal{G}} = \{e \in \mathcal{E}^{\mathbb{Z}} : \forall i \in \mathbb{Z}, t(e_i) = i(e_{i+1})\}$$

Étant donné un sous-shift  $\mathbf{T} \subset \mathcal{A}^{\mathbb{Z}}$ , on considère le *graphe du langage d’ordre  $n$*  appelé aussi le *graphe de Rauzy d’ordre  $n$*   $\mathcal{G}_{\mathbf{T}}^n = (\mathcal{V}_{\mathbf{T}}^n, \mathcal{E}_{\mathbf{T}}^n)$  où  $\mathcal{V}_{\mathbf{T}}^n = \mathcal{L}_{n-1}(\mathbf{T})$  et  $(u_0 \dots u_{n-2}, v_0 \dots v_{n-2}) \in \mathcal{E}_{\mathbf{T}}^n$  si et seulement si  $u_0 \dots u_{n-2} v_{n-2} = u_0 v_0 \dots v_{n-2} \in \mathcal{L}_n(\mathbf{T})$ .

**Proposition 2.2.3** ([LM95]). *En dimension 1, un sous shift de type fini  $\mathbf{T}(\mathcal{A}, 1, \mathcal{F})$  où  $\mathcal{F} \subset \mathcal{A}^n$  est conjugué au sous-shift de type fini associé au graphe de Rauzy  $\mathcal{G}_{\mathbf{T}}^n$ .*

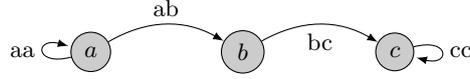
La conjugaison est établit par la fonction  $\varphi_n : \mathbf{T}_{\mathcal{G}_{\mathbf{T}}^n} \rightarrow \mathbf{T}$  qui à une arête renvoi la première lettre de l’étiquette associée.

Exemple 2.2.5. Si on considère  $\mathbf{T}(\{0, 1\}, 1, \{11\})$ , on a :



le graphe du langage d’ordre 2    le graphe du langage d’ordre 3

*Exemple 2.2.6.* Si on considère  $\mathbf{T}(\{a, b, c\}, 1, \{bb, ba, cb, ac, ca\})$ , le graphe du langage d'ordre 2 est :



**Corollaire 2.2.4.** *En dimension 1 il est décidable de savoir si un sous-shift de type fini est vide.*

Il suffit d'écrire le graphe du langage d'ordre  $n$  où  $n$  est la taille maximale des motifs interdits et de vérifier qu'il y ait un chemin infini.

### Sous-shift de type fini et conjugaisons

**Proposition 2.2.5.** *Si un sous-shift  $\mathbf{T} \subset \mathcal{A}^{\mathbb{Z}^d}$  est conjugué à un sous-shift de type fini  $\mathbf{T}(\mathcal{B}, d, \mathcal{F})$  alors  $\mathbf{T}$  est un sous-shift de type fini.*

*Démonstration.* Soit  $\varphi : \mathbf{T} \rightarrow \mathbf{T}(\mathcal{B}, d, \mathcal{F})$  le morphisme réalisant la conjugaison. On considère  $\mathcal{F}' = \{\varphi^{-1}(p) : p \in \mathcal{F}\}$  un ensemble de motif  $d$ -dimensionnel sur  $\mathcal{A}$ . On veut montrer que  $\mathbf{T} = \mathbf{T}(\mathcal{A}, d, \mathcal{F}')$ . On a :

$$x \in \mathbf{T}(\mathcal{A}, d, \mathcal{F}') \iff \begin{array}{c} \varphi(x) \in \mathbf{T}(\mathcal{B}, d, \mathcal{F}) \\ \text{car aucun motif de } \mathcal{F} \text{ n'apparaît dans } \varphi(x) \end{array} \iff \begin{array}{c} x \in \mathbf{T} \\ \text{car } \varphi \text{ est une conjugaison} \end{array}$$

□

**Proposition 2.2.6.** *Tout sous-shift de type fini  $\mathbf{T}(\mathcal{A}, d, \mathcal{F})$  est topologiquement conjugué à :*

- un sous-shift de type fini  $\mathbf{T}(\mathcal{B}_1, d, \mathcal{F}_1)$  où  $\mathcal{F}_1 \subset \mathcal{B}^{[0,1]^d}$ ,
- un sous-shift de type fini  $\mathbf{T}(\mathcal{B}_2, d, \mathcal{F}_2)$  formé de tuiles de Wang,

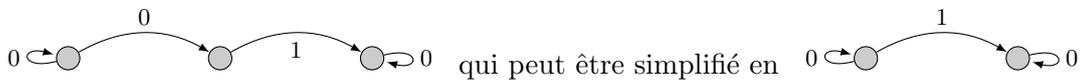
### 2.2.3 Sous-shift sofique

#### Définition

Un sous-shift  $\mathbf{T} \subset \mathcal{B}^{\mathbb{Z}^d}$  est *sofique* s'il existe un sous-shift de type fini  $\mathbf{T}(\mathcal{A}, d, \mathcal{F})$  et un facteur  $\pi : \mathcal{A}^{\mathbb{Z}^d} \rightarrow \mathcal{B}^{\mathbb{Z}^d}$  tel que  $\mathbf{T} = \pi(\mathbf{T}(\mathcal{A}, d, \mathcal{F}))$ .

*Exemple 2.2.7.* Considérons le sous-shift  $\mathbf{T} = \{x \in \{0, 1\}^{\mathbb{Z}^d} : \text{il y a au plus un } i \in \mathbb{Z}^d \text{ tel que } x_i = 1\}$ . Le sous-shift  $\mathbf{T}$  est sofique mais pas de type fini. En dimension 1, il suffit de prendre le sous-shift de type fini de l'exemple 2.2.6 et de prendre le facteur  $\pi : \{a, c\} \mapsto 0$  et  $\pi : b \mapsto 1$ . En dimension supérieure on généralise en prenant un produit de tels sous-shifts.

On peut représenter ce sous shift sous la forme d'un graphe à partir de l'exemple 2.2.6.



L'ensemble des sous-shifts sofiques est dénombrable ce qui n'est pas le cas de l'ensemble des sous-shifts donc il existe des sous-shifts qui ne sont pas sofiques. Les exemples suivant en exhibent :

*Exemple 2.2.8.* Le sous-shift  $\mathbf{T}(\{0, a, b\}, 1, \{0b, a0, 0a^n b^m 0 : n \neq m\})$  n'est pas sofique.

*Exemple 2.2.9.* Considérons le sous-shift

$$\mathbf{T} = \left\{ x \in \{0, 1, 2\}^{\mathbb{Z}^2} : \begin{array}{l} \text{si } x_{(i_0, j_0)} = 2 \text{ alors } x_{(i_0, j)} = 2 \text{ pour tout } j \in \mathbb{Z} \\ \text{pour } i \neq i_0 \text{ on a } x_{(i, j)} \neq 2 \text{ et } x_{(i, j)} = x_{(i, j_0 - j)} \text{ pour tout } j \in \mathbb{Z} \end{array} \right\}.$$

Le sous-shift  $\mathbf{T}$  n'est pas sofique.

**Proposition 2.2.7.** *Pour tout sofique  $\mathbf{T} \subset \mathcal{B}^{\mathbb{Z}^d}$  il existe un sous-shift de type fini  $\mathbf{T}(\mathcal{A}, d, \mathcal{F})$  et un facteur  $\pi : \mathcal{A}^{\mathbb{Z}^d} \rightarrow \mathcal{B}^{\mathbb{Z}^d}$  lettre à lettre (c'est à dire que  $\bar{\pi} : \mathcal{A} \rightarrow \mathcal{B}$ ) tel que  $\mathbf{T} = \pi(\mathbf{T}(\mathcal{A}, d, \mathcal{F}))$ .*

### Cas de la dimension 1

Soit  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  un graphe orienté où les arêtes sont étiquetées par  $\pi : \mathcal{E} \rightarrow \mathcal{A}$ . Alors  $\pi(\mathbf{T}_{\mathcal{G}})$  est un sous-shift sofique et réciproquement tout sofique peut s'écrire de cette manière. On peut supposer qu'il n'y a pas deux arêtes partant d'un sommet  $v \in \mathcal{V}$  étiquetées par la même valeur. Dans ce cas on dit que  $\mathcal{G}$  est *résolvant à droite*.

**Théorème 2.2.8** ([Wei73, LM95]). *En dimension 1, le langage d'un sofique est rationnel, c'est à dire reconnu par un automate fini, c'est à dire il correspond aux motifs obtenus en parcourant un graphe étiqueté.*

## 2.3 Quelques invariants topologiques

### 2.3.1 Complexité et entropie

Étant donné un sous-shift  $\mathbf{T}$ , on définit la *fonction complexité* de  $\mathbf{T} \subset \mathcal{A}^{\mathbb{Z}^d}$  comme la fonction

$$p_{\mathbf{T}} : \mathbb{N} \rightarrow \mathbb{N} \\ n \mapsto \text{Card} \{u \in \mathcal{A}^{[0, n-1]^d} : \exists x \in \mathbf{T} \text{ tel que } u \sqsubset x\} = \text{Card}(\mathcal{L}_n(\mathbf{T})).$$

**Théorème 2.3.1.** *Si  $\mathbf{T} \subset \mathcal{A}^{\mathbb{Z}}$  est un sous-shift unidimensionnel infini alors  $p_{\mathbf{T}}(n) \geq n + 1$ .*

*Démonstration.* Supposons que  $\mathbf{T} \subset \mathcal{A}^{\mathbb{Z}}$  est infini, l'alphabet le définissant a au moins deux lettres donc  $p_{\mathbf{T}}(1) \geq 2$ . Supposons qu'il existe  $n \geq 1$  tel que  $p_{\mathbf{T}}(n+1) = p_{\mathbf{T}}(n)$ . Pour tout  $u \in \mathcal{L}_{n-1}(\mathbf{T})$ , il existe  $a$  tel que  $ua \in \mathcal{L}_n(\mathbf{T})$  et comme  $p_{\mathbf{T}}(n+1) = p_{\mathbf{T}}(n)$ , ce  $a$  est unique et donc  $u$  a une unique extension. Ainsi pour tout  $u \in \mathcal{L}_{n-1}(\mathbf{T})$ , il existe une unique extension  $x \in \mathbf{T}$  et donc le cardinal de  $\mathbf{T}$  est  $p_{\mathbf{T}}(n)$ .  $\square$

**Conjecture 1.** *En dimension supérieure, le problème similaire qui consiste à dire que*

*"Card  $\{u \in \mathcal{A}^{[0, n-1] \times [0, m-1]} : \exists x \in \mathbf{T} \text{ tel que } u \sqsubset x\} \leq nm$  si et seulement si  $x$  est périodique"*

*reste ouvert et s'appelle la conjecture de Niva.*

La fonction complexité  $p_{\mathbf{T}}$  n'est pas un invariant topologique, par exemple deux sous-shifts  $\mathbf{T}$  et  $\mathbf{T}'$  conjugués n'ont pas forcément le même alphabet d'où  $p_{\mathbf{T}}(1) \neq p_{\mathbf{T}'}(1)$ .

Un  $d$ -cube de côté  $nk$  peut être divisé en  $k^d$   $d$ -cubes de côtés  $n$ , on a donc  $p_{\mathbf{T}}(nk) \leq p_{\mathbf{T}}(n)^{k^d}$ . On en déduit que  $\frac{\log(p_{\mathbf{T}}(nk))}{(kn)^d} \leq \frac{\log(p_{\mathbf{T}}(n))}{n^d}$ . Ainsi pour tout  $m \in \mathbb{N}$ , on fait la division euclidienne  $m = nk + m'$  où  $m' \in [0, n[$  et on obtient :

$$\frac{\log(p_{\mathbf{T}}((n+1)k))}{(nk+m')^d} \geq \frac{\log(p_{\mathbf{T}}(m))}{m^d} \geq \frac{\log(p_{\mathbf{T}}(kn))}{(nk+m')^d} \text{ ce qui donne } \limsup_{m \rightarrow \infty} \frac{\log(p_{\mathbf{T}}(m))}{m^d} = \limsup_{n \rightarrow \infty} \frac{\log(p_{\mathbf{T}}(kn))}{(kn)^d}.$$

Ainsi pour tout  $n \in \mathbb{N}$  on a  $\limsup_{n \rightarrow \infty} \frac{\log(p_{\mathbf{T}}(kn))}{(kn)^d} \leq \frac{\log(p_{\mathbf{T}}(n))}{n^d}$ . On en déduit que la limite de  $n \mapsto \frac{\log(p_{\mathbf{T}}(n))}{n^d}$  lorsque  $n \rightarrow \infty$  existe. On définit alors l'*entropie topologique* :

$$h(\mathbf{T}) = \lim_{n \rightarrow \infty} \frac{\log(p_{\mathbf{T}}(n))}{n^d} = \inf_{n \in \mathbb{N}} \frac{\log(p_{\mathbf{T}}(n))}{n^d}.$$

On a  $0 \leq h(\mathbf{T}) \leq \log(\mathcal{A}) = h(\mathcal{A}^{\mathbb{Z}^d})$ . De plus, comme le dit la proposition suivante, l'entropie est un invariant topologique.

**Proposition 2.3.2.** Soient  $\mathbf{T} \subset \mathcal{A}^{\mathbb{Z}^d}$  et  $\mathbf{T}' \subset \mathcal{B}^{\mathbb{Z}^d}$ . Si  $\pi : \mathbf{T} \rightarrow \mathbf{T}'$  est un facteur alors  $h(\mathbf{T}) \geq h(\mathbf{T}')$ .

*Démonstration.* Le facteur  $\pi$  est défini par bloc par la fonction locale  $\bar{\pi} : \mathcal{A}^{[-r,r]} \rightarrow \mathcal{B}$ . Pour tout  $u' \in \mathcal{L}_n(\mathbf{T}')$ , alors il existe  $u \in \mathcal{A}^{[-r,r+n-1]}$  un motif apparaissant dans  $\mathbf{T}$  tel que  $\bar{\pi}(u) = u'$ . Ainsi

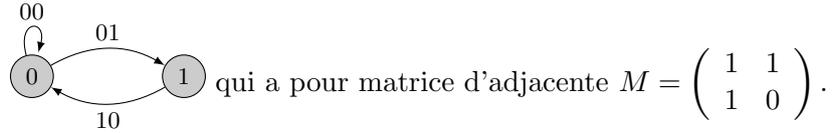
$$\begin{aligned} p_{\mathbf{T}'}(n) &= \text{Card} \left\{ u \in \mathcal{A}^{[0,n-1]^d} : \exists x \in \mathbf{T} \text{ tel que } u \sqsubset x \right\} \\ &\leq \sum_{u \in \mathcal{L}_n(\mathbf{T}')} \text{Card}(\bar{\pi}^{-1}(u) \cap \mathbf{T}) \\ &\leq \text{Card} \left\{ u \in \mathcal{A}^{[0,2r+n-1]^d} : \exists x \in \mathbf{T} \text{ tel que } u \sqsubset x \right\} \\ &\leq p_{\mathbf{T}}(2r+n) \end{aligned}$$

On a donc  $\frac{p_{\mathbf{T}'}(n)}{n^d} \leq \frac{p_{\mathbf{T}}(2r+n)}{n^d} = \frac{(2r+n)^d}{n^d} \frac{p_{\mathbf{T}}(2r+n)}{(2r+n)^d}$  et par passage à la limite on obtient  $h(\mathbf{T}') \leq h(\mathbf{T})$ .  $\square$

**Corollaire 2.3.3.** Si  $\mathbf{T}$  et  $\mathbf{T}'$  sont topologiquement conjugués alors  $h(\mathbf{T}) = h(\mathbf{T}')$ .

*Exemple 2.3.1.* Les full-shifts  $(\mathcal{A}^{\mathbb{Z}^d}, \sigma)$  et  $(\mathcal{B}^{\mathbb{Z}^d}, \sigma)$  sont topologiquement conjugués si  $\text{Card}(\mathcal{A}) = \text{Card}(\mathcal{B})$ .

*Exemple 2.3.2.* On considère le sous-shift *pair*  $\mathbf{T}_{\text{even}} = \mathbf{T}(\{0,1\}, 1, \mathcal{F})$  où  $\mathcal{F} = \{11\}$ . On le peut le définir par le graphe suivant :



On vérifie que le nombre de chemins qui vont du sommet  $i$  au sommet  $j$  en utilisant  $n$  arêtes est le coefficient  $(i, j)$  de la matrice  $M^n$ . On en déduit que le nombre de mots de longueur  $n$  est

$$\text{Card}(\mathcal{L}_n(\mathbf{T}_{\text{even}})) = \sum_{i,j \in \{0,1\}} M_{i,j}^n.$$

Par le théorème de Perron-Frobenius,  $M$  admet une valeur propre dominante noté  $\lambda = \max\{|\mu| : \mu \text{ valeur propre de } M\}$  associé à un vecteur propre positif  $v$ . Soit  $c = \min(v_0, \dots, v_r)$  et  $d = \max(v_0, \dots, v_r)$ , on a donc  $\frac{c}{d} \lambda^n \leq \sum_{i,j \in \{0,1\}} M_{i,j}^n \leq \frac{rd}{c} \lambda^n$ . Ainsi, seule la valeur propre dominante compte dans le calcul de l'entropie, on a donc  $h(\mathbf{T}_{\text{even}}) = \log\left(\frac{1+\sqrt{5}}{2}\right)$ .

La méthode de l'exemple précédent peut être généralisé pour obtenir le théorème suivant :

**Théorème 2.3.4** ([LM95]). On considère  $\mathbf{T}$  est un sous-shift sofique qui s'écrit sous la forme  $\pi(\mathbf{T}_{\mathcal{G}})$  où  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  est un graphe orienté dont les arêtes sont étiquetées par  $\pi : \mathcal{E} \rightarrow \mathcal{A}$  et qui est résolvant à droite. Étant donné la matrice  $M$  d'adjacence associée au graphe  $\mathcal{G}$  on a  $h(\mathbf{T}) = \log(\max\{|\lambda| : \lambda \text{ valeur propre de } M\})$ .

### 2.3.2 Orbite, transitivité et minimalité

Pour  $x \in \mathcal{A}^{\mathbb{Z}^d}$  on définit l'*orbite* de  $x$  par  $\mathcal{O}(x) = \{\sigma^i(x) : i \in \mathbb{Z}^d\}$ . L'orbite n'est pas nécessairement fini, on considère donc la fermeture  $\overline{\mathcal{O}(x)}$ . Le fermé  $\overline{\mathcal{O}(x)}$  est  $\sigma$ -invariant, c'est donc un sous-shift. On remarque que  $y \in \mathcal{O}(x)$  si et seulement si  $\mathcal{O}(y) \subset \mathcal{O}(x)$ .

Un sous-shift  $\mathbf{T}$  est *transitif* si pour tout ouvert  $\mathcal{U}$  et  $\mathcal{V}$  de  $\mathbf{T}$  il existe  $x \in \mathbf{T}$  et  $i \in \mathbb{Z}^d$  tel que  $\mathcal{U}$  et  $\sigma^n(x) \in \mathcal{V}$ . Autrement dit pour  $u, v \in \mathcal{L}(\mathbf{T})$  il existe  $x \in \mathbf{T}$  tel que  $u, v \in \mathcal{L}(x)$ . La transitivité est conservée par conjugaison.

**Proposition 2.3.5.** *Un sous-shift  $\mathbf{T}$  est transitif si et seulement si il existe  $x \in \mathbf{T}$  tel que  $\mathbf{T} = \overline{\mathcal{O}(x)}$ .*

*Démonstration.* Pour toute configuration  $x \in \mathcal{A}^{\mathbb{Z}^d}$ ,  $\mathcal{O}(x)$  est transitif, en effet tout motif qui apparaît dans  $\mathcal{O}(x)$  apparaît dans  $x$ .

Supposons que  $\mathbf{T}$  soit un sous-shift transitif. Soit  $(u_i)_{i \in \mathbb{N}}$  une énumération de motifs de  $\mathcal{L}(\mathbf{T})$ . Par transitivité, il existe  $n_0, n_1 \in \mathbb{Z}^d$  tels que  $V_1 = [u_0] \cap [u_1] \cap \mathbf{T} \neq \emptyset$ . Pour tout  $k \geq 1$ , on construit ainsi par récurrence  $n_0, \dots, n_k \in \mathbb{Z}^d$  tels que  $V_k = [u_0] \cap \dots \cap [u_k] \cap \mathbf{T} \neq \emptyset$ . On a une suite décroissante de fermés et par compacité, il existe  $x \in \bigcap_{k \in \mathbb{N}} V_k$ . Ainsi  $x \in \mathbf{T}$  contient tous les motifs de  $\mathcal{L}(\mathbf{T})$  d'où  $\mathbf{T} = \mathcal{O}(x)$ .  $\square$

Un sous-shift  $\mathbf{T}$  est *minimal*, s'il ne contient pas de sous-shift non vide strictement inclus dans  $\mathbf{T}$ .

**Proposition 2.3.6.** *Soit  $\mathbf{T}$  un sous-shift. Les propriétés suivantes sont équivalentes :*

1.  $\mathbf{T}$  est minimal ;
2. pour tout  $x \in \mathbf{T}$ ,  $\mathcal{O}(x)$  est dense dans  $\mathbf{T}$  ;
3. pour tous  $x, y \in \mathbf{T}$ ,  $\mathcal{L}(x) = \mathcal{L}(y)$ .

*Démonstration.* 1.  $\Rightarrow$  2. Pour tout  $x \in \mathbf{T}$ ,  $\overline{\mathcal{O}(x)} \subset \mathbf{T}$  est un sous-shift donc  $\overline{\mathcal{O}(x)} = \mathbf{T}$ .

2.  $\Rightarrow$  1. Si  $\mathbf{T}$  est non minimal alors il existe un sous-shift  $\mathbf{T}' \subsetneq \mathbf{T}$ . Ainsi pour tout  $x \in \mathbf{T}'$ , son orbite n'est pas dense dans  $\mathbf{T}$ .

2.  $\Leftrightarrow$  3. Pour toutes configurations  $x, y \in \mathbf{T}$ , on a  $\overline{\mathcal{O}(x)} = \overline{\mathcal{O}(y)}$  si et seulement si  $\mathcal{L}(x) = \mathcal{L}(y)$ .  $\square$

**Proposition 2.3.7.** *Les notions de minimalité et de transitivité sont des invariants topologiques.*

**Théorème 2.3.8.** *Tout sous-shift contient un sous-shift minimal.*

*Démonstration.* Pour tout ouvert  $U \subset \mathcal{A}^{\mathbb{Z}^d}$ , on note  $\mathcal{O}(U) = \bigcup_{n \in \mathbb{Z}^d} \sigma^n(U)$ , c'est donc un ouvert comme union d'ouverts. On considère  $(U_i)_{i \in \mathbb{N}}$  une base dénombrable de la topologie, par exemple une énumération des cylindres.

On pose  $F_0 = \mathbf{T}$  et pour  $k \geq 1$  on définit par récurrence  $F_k = \begin{cases} F_{k-1} & \text{si } F_{k-1} \subset \mathcal{O}(U_k) \\ F_{k-1} \setminus \mathcal{O}(U_k) & \text{sinon} \end{cases}$ .

Par construction  $(F_k)_{k \in \mathbb{N}}$  forme une suite décroissante de fermés non vides  $\sigma$ -invariants. Par compacité  $F = \bigcap_{k \in \mathbb{N}} F_k$  est donc un fermé  $\sigma$ -invariant. On va montrer que  $F$  est minimal.

Soient  $x, y \in F$  tels que  $\mathcal{L}(x) \setminus \mathcal{L}(y) \neq \emptyset$ . Il existe donc un ouvert  $U_i$  tel que  $x \in \mathcal{O}(U_i)$  et  $y \notin \mathcal{O}(U_i)$ . Comme  $F \subset F_{i-1}$ , ce n'est pas un sous ensemble de  $\mathcal{O}(U_i)$  donc  $F_i = F_{i-1} \setminus \mathcal{O}(U_i)$  ce qui est incompatible avec le fait que  $x \in F$ .  $\square$

### 2.3.3 Périodicité et propriété de récurrence

Une configuration  $x \in \mathcal{A}^{\mathbb{Z}^d}$  est *périodique suivant la direction*  $j \in \mathbb{Z}^d \setminus \{0\}$  si  $\sigma^j(x) = x$ ,  $j$  est une période de  $x$ . Une configuration est *périodique* si elle est périodique suivant  $d$  directions linéairement indépendantes.

Soient  $\mathbf{T}$  et  $\mathbf{T}'$  deux sous-shifts conjugués. Il y a une correspondance entre les orbites périodique de  $\mathbf{T}$  et  $\mathbf{T}'$ . On voit par exemple que  $(\mathcal{A}^{\mathbb{Z}}, \sigma)$  et  $(\mathcal{B}^{\mathbb{Z}}, \sigma)$  ne sont pas conjugués si  $\mathcal{A}$  et  $\mathcal{B}$  n'ont pas le même nombre d'éléments.

**Proposition 2.3.9.** *Soit  $x \in \mathcal{A}^{\mathbb{Z}^d}$ , les propriétés suivantes sont équivalentes :*

1.  $x$  est périodique ;
2.  $\mathcal{O}(x)$  est fini ;
3.  $\mathcal{O}(x)$  est fermé.

*Démonstration.* On a facilement  $1. \iff 2. \implies 3.$

On suppose que  $\mathcal{O}(x)$  est infini. On construit une suite  $(\sigma^{n_i}(x))_{i \in \mathbb{N}}$  telle que pour tout  $i \in \mathbb{N}$ , il existe  $k_i$  tel que  $\sigma^{n_i}(x) \notin \bigcup_{|j| \leq i} [\sigma^j(x)]_{[-k_i, k_i]^d}$ . Par compacité on construit un  $y$  obtenu comme limite d'éléments de  $\mathcal{O}(x)$  mais qui n'est pas dans  $\mathcal{O}(x)$ .  $\square$

**Proposition 2.3.10.** *Si un sous-shift de type fini  $\mathbf{T} \subset \mathcal{A}^{\mathbb{Z}^d}$  contient une configuration périodiques suivant  $d - 1$  directions non linéairement indépendantes alors il contient une configuration périodique.*

Il existe différentes propriétés plus faibles que la périodicité pour décrire la récurrence :

Une configuration  $x \in \mathcal{A}^{\mathbb{Z}^d}$  est *quasi-périodique* si pour tout ouvert  $\mathcal{U}$  contenant  $x$  il existe  $a_1, \dots, a_d \in \mathbb{Z}^d$  linéairement indépendants tels que pour tous  $i_1, \dots, i_d \in \mathbb{Z}$  on ait  $\sigma^{i_1 a_1 + \dots + i_d a_d}(x) \in \mathcal{U}$ . Autrement dit, tout motif est répété périodiquement, mais pas forcément la même période pour deux motifs donnés. On dit que  $x \in \mathcal{A}^{\mathbb{Z}^d}$  est *quasi-périodique faible* s'il existe  $a_1, \dots, a_d \in \mathbb{Z}^d$  linéairement indépendants et  $c \in \mathbb{Z}^d$  tels que pour tous  $i_1, \dots, i_d \in \mathbb{Z}$  on ait  $\sigma^{i_1 a_1 + \dots + i_d a_d + c}(x) \in \mathcal{U}$ .

Une configuration  $x \in \mathcal{A}^{\mathbb{Z}^d}$  est *uniformément récurrente* si pour tout ouvert  $\mathcal{U}$  contenant  $x$  il existe  $\mathbb{U} \subset \mathbb{Z}^d$  fini tel que pour tout  $i \in \mathbb{Z}^d$ , il existe  $j \in \mathbb{U}$  tel que  $\sigma^{i+j}(x) \in \mathcal{U}$ . Autrement dit, pour tout motif  $u$  il existe  $n \in \mathbb{N}$  tel que pour tout  $i \in \mathbb{Z}^d$  on ait  $u \sqsubset x_{i+\mathbb{U}_n}$ .

Une configuration  $x \in \mathcal{A}^{\mathbb{Z}^d}$  est *linéairement récurrente* si il existe une constante  $K$  telle que pour tout motif  $u \in \mathbb{U}_n$  et pour tout  $i \in \mathbb{Z}^d$  on ait  $u \sqsubset x_{i+\mathbb{U}_{Kn}}$ .

Une configuration  $x \in \mathcal{A}^{\mathbb{Z}^d}$  est *récurrente* si pour tout ouvert  $\mathcal{U}$  contenant  $x$  il existe  $i \in \mathbb{Z}^d \setminus \{0\}$  tel que  $\sigma^i(x) \in \mathcal{U}$ .

Un sous-shift  $\mathbf{T}$  est *périodique* (respectivement *quasi-périodique*, *quasi-périodique faible*, *linéairement récurrent*, *uniformément récurrent*, *récurrent*) si il existe une configuration  $x$  périodique (respectivement quasi-périodique, quasi-périodique faible, uniformément récurrent, récurrent) telle que  $\overline{\mathcal{O}(x)} = \mathbf{T}$ .

*Remarque.* Étant donné une configuration ou un sous-shift, on a bien sûr :

$$\begin{aligned} \text{Périodique} &\implies \text{Quasi-périodique} \implies \text{Quasi-périodique faible} \implies \text{Linéairement récurrent} \\ &\implies \text{Uniformément récurrent} \implies \text{Récurrent} \end{aligned}$$

**Proposition 2.3.11.** *Les notions de périodique, quasi-périodique, quasi-périodique faible, linéairement récurrent, uniformément récurrent et récurrent sont des invariants topologiques.*

**Proposition 2.3.12.** *Une configuration  $x \in \mathcal{A}^{\mathbb{Z}^d}$  est récurrente si et seulement si pour tout ouvert  $\mathcal{U}$  contenant  $x$  il existe  $\mathbb{U} \subset \mathbb{Z}^d$  infini tel que  $\sigma^i(x) \in \mathcal{U}$ .*

*Démonstration.* Soient  $x \in \mathcal{A}^{\mathbb{Z}^d}$  récurrent et  $\mathcal{U}$  un ouvert. On considère  $\mathbb{U} \subset \mathbb{Z}^d$  tel que  $\sigma^i(x) \in \mathcal{U}$ . Supposons que  $\mathbb{U}$  soit fini alors  $\mathcal{V} = \bigcap_{i \in \mathbb{U}} \sigma^{-i}(\mathcal{U})$  est un ouvert contenant  $x$ . Par récurrence de  $x$ , il existe  $j \neq 0$  tel que  $\sigma^j(x) \in \mathcal{V}$ , d'où  $\sigma^{i+j}(x) \in \mathcal{U}$  pour tout  $i \in \mathbb{U}$ . Ainsi  $j + \mathbb{U} = \mathbb{U}$  ce qui est impossible si  $\mathbb{U}$  est fini.  $\square$

**Théorème 2.3.13** (Gottschalk). *Le sous-shift  $\overline{\mathcal{O}(x)}$  est minimal si et seulement si  $x$  est uniformément récurrent.*

*Démonstration.* Soient  $x$  uniformément récurrent et  $y \in \overline{\mathcal{O}(x)}$  d'où  $\mathcal{L}(y) \subset \mathcal{L}(x)$ . Soit  $\mathbb{U} \subset \mathbb{Z}^d$  fini et  $u \in \mathcal{L}_{\mathbb{U}}(x)$ , il existe  $n$  tel que pour tout  $i \in \mathbb{Z}^d$  on ait  $u \sqsubset x_{i+[0,n-1]^d}$ . Comme  $y_{[0,n-1]^d} \in \mathcal{L}(x)$ , on en déduit que  $u \in \mathcal{L}(y)$  d'où le résultat.

Réciproquement on suppose que  $x$  n'est pas uniformément récurrent. Donc il existe  $\mathbb{U} \subset \mathbb{Z}^d$  fini tel pour tout sous ensemble fini  $\mathbb{U}' \subset \mathbb{Z}^d$ , il existe  $i \in \mathbb{Z}^d$  tel que  $\sigma^{i+j}(x) \notin [x_{\mathbb{U}}]$  pour tout  $j \in \mathbb{U}'$ . Soit  $(j_n)_{n \in \mathbb{N}}$  une énumération de  $\mathbb{Z}^d$ . On construit une suite  $(i_n)_{n \in \mathbb{N}} \in (\mathbb{Z}^d)^{\mathbb{N}}$  telle que pour tout  $n \in \mathbb{N}$ , on ait  $\sigma^{i_n+j_{n'}}(x) \notin [x_{\mathbb{U}}]$  pour  $n' \leq n$ . Si  $y$  est une valeur d'adhérence de la suite  $(\sigma^{i_n}(x))_{n \in \mathbb{N}}$  alors, par construction,  $\sigma^k(y) \notin [x_{\mathbb{U}}]$  pour tout  $k \in \mathbb{Z}^d$ .  $\square$

**Théorème 2.3.14.** *Un sous-shift minimal est soit fini, soit non dénombrable.*

*Démonstration.* Soit  $\mathbf{T} = \{x_0, x_1, x_2, \dots\}$  un sous-shift minimal dénombrable. On veut montrer qu'il existe un cylindre  $U$  tel que  $U$  contient exactement un seul élément. Supposons le contraire, c'est à dire que pour tout cylindre  $U$ , on a  $U \cap \mathbf{T} = \emptyset$  ou  $|U \cap \mathbf{T}| \geq 2$ . On va alors construire une chaîne d'ouvert-fermés  $F_1 \subset F_2 \subset \dots$  telle que  $x_i \in F_i$  pour tout  $i \in \mathbb{N}$  mais  $F_i \neq \mathbf{T}$ . On obtiendra alors une contradiction par compacité.

On pose  $F_0 = \emptyset$ . On suppose que  $F_{i-1}$  est construit. Si  $x_i \in F_{i-1}$  alors  $F_i = F_{i-1}$ . Sinon, il existe un cylindre  $V$  tel que  $x_i \in V$  et  $V \cap F_{i-1} = \emptyset$ . Or  $V$  contient un élément  $x_k \in \mathbf{T}$  avec  $k > i$ . Il existe donc un cylindre  $V'$  tel  $x_i \in V'$  et  $x_k \notin V'$ . On pose  $F_i = F_{i-1} \cup V'$ .

Il existe donc un cylindre  $U$  qui contient un seul élément  $x \in \mathbf{T}$ . Comme  $\mathbf{T}$  est minimal,  $x$  est uniformément récurrent. Il existe une infinité de  $n \in \mathbb{Z}^d$  tel que  $\sigma^n(x) \in U$  d'où  $\sigma^n(x) = x$ . On en déduit que  $x$  est périodique.  $\square$

**Corollaire 2.3.15.** *Si un sous-shift est non vide alors il admet une configuration uniformément récurrente. Si de plus cette configuration n'est pas périodique alors le sous-shift n'est pas dénombrable.*

## Chapitre 3

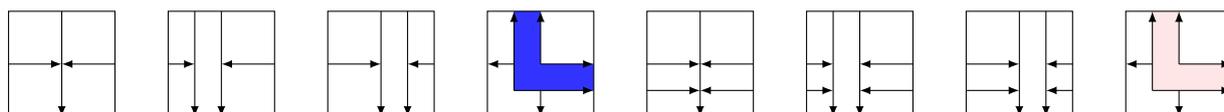
# Pavages apériodiques

Wang proposa la conjecture suivante [Wan61] : “Étant donné un jeu de tuile de Wang qui définit un espace de pavage non vide, alors il existe nécessairement une configuration périodiques”. On rappelle que périodique signifie que la configuration est périodique suivant  $d$  directions indépendantes ou  $d$  est la dimension de l’espace à paver. Depuis l’infirmité de cette conjecture par R. Berger [Ber66] qui a exhibé un jeu de tuiles forçant les configurations construites à partir de celui ci à être apériodique, il y a un défi à trouver un jeu de tuiles ayant cette propriété qui utilise le moins de tuiles possible. Voilà un petit historique des pavages apériodiques, on compte les tuiles suivant si on les regarde à translation près, à isométrie près ou si on considère le pavage par tuile de Wang associé (si il existe). E. Jeandel (2010) a montré par une méthode exhaustive qu’il ne peut pas y avoir un jeu de tuiles de Wang à 9 tuiles forçant les configurations à être apériodique.

Auteur	Nombre de tuiles de Wang	Nombre de tuiles à translation près	Nombres de tuile à isométrie près
R. Berger [Ber66]	20 426		
R. Berger (1966)	104		
D. E. Knuth (1966)	92		
R. Penrose[Pen78]		20	2
R. M. Robinson [Rob71]	56	18	6
R. Ammann (1978)[AGS92]		16	2
J. Kari [Kar96]	14		
K. Culick [II96]	13		

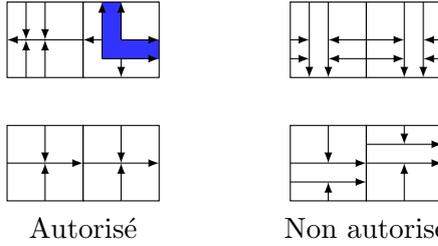
### 3.1 Pavage de Robinson

Il existe différentes façons de définir le pavage de Robinson, on considère le jeu de tuiles suivant, noté *Robi*, modulo les rotations d’angle  $\frac{\pi}{2}$ ,  $\pi$  et  $-\frac{\pi}{2}$  :

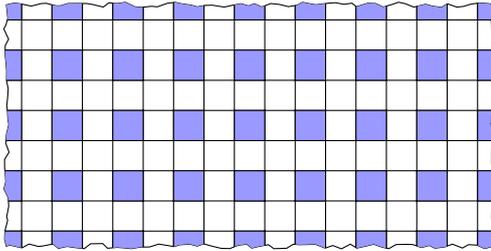


On considère l’ensemble des motifs interdits  $\mathcal{F}_{\text{Robi}}$  :

- qui force la correspondance entre les flèches entrantes et sortantes de deux tuiles adjacentes :



– auquel on rajoute les motifs interdits qui imposent de mettre une tuile de  $\left\{ \begin{matrix} \text{L-tile} \\ \text{rotated L-tile} \\ \text{rotated L-tile} \\ \text{rotated L-tile} \end{matrix} \right\}$  suivant le réseau suivant :

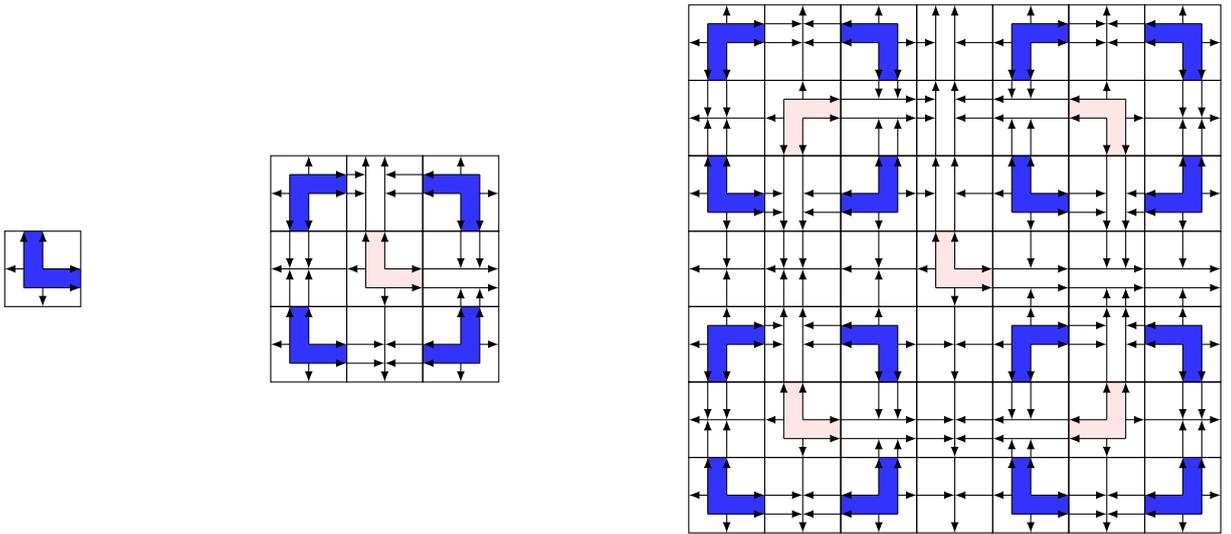


où  $\square \in \left\{ \begin{matrix} \text{L-tile} \\ \text{rotated L-tile} \\ \text{rotated L-tile} \\ \text{rotated L-tile} \end{matrix} \right\}$

On définit le sous-shift de type fini de Robinson par  $\mathbf{T}_{Robi} = \mathbf{T}(\mathcal{R}_{Robi}, 2, \mathcal{F}_{Robi})$ .

**Théorème 3.1.1** ([Rob71]). *Le sous-shift de type fini  $\mathbf{T}_{Robi}$  est non vide et formé uniquement de configurations aperiodiques.*

*Démonstration.* Par récurrence, on construit des super-tuiles d'ordre  $n$  qui correspondent à un motif élémentaire de taille  $2^{n+1} - 1$  qui a les mêmes propriétés que la tuile .



Super tuile d'ordre 0

Super tuile d'ordre 1

Super tuile d'ordre 2

FIGURE 3.1 – Super tuile d'ordre 0, 1 et 2.

Par compacité on en déduit que  $\mathbf{T}_{Robi} \neq \emptyset$ .

Pour montrer l'aperiodicité, on va montrer que toute configuration  $x \in \mathbf{T}_{Robi}$  contient une super-tuile d'ordre  $n$  pour tout  $n \in \mathbb{N}$ . Ainsi, comme le centre de deux tuiles sont forcément espacés de  $2^n$  (voir figure 3.2) on en déduit que la configuration n'est pas périodique.

Pour montrer que toute configuration admet une super tuile d'ordre  $n$ , on montre que la

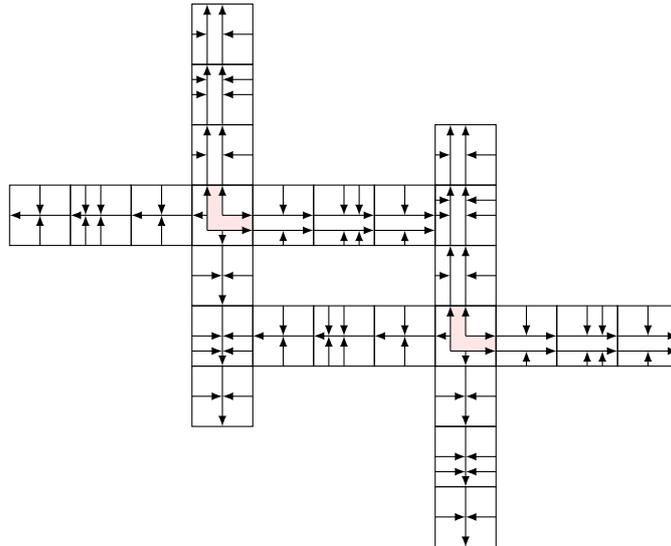
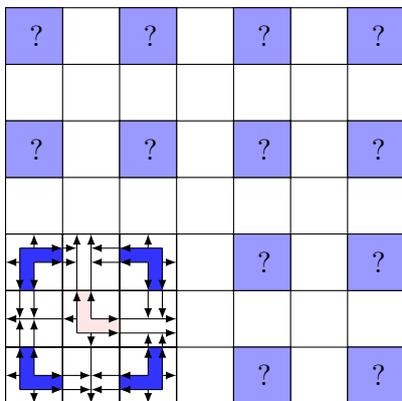
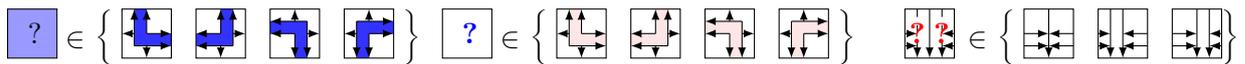
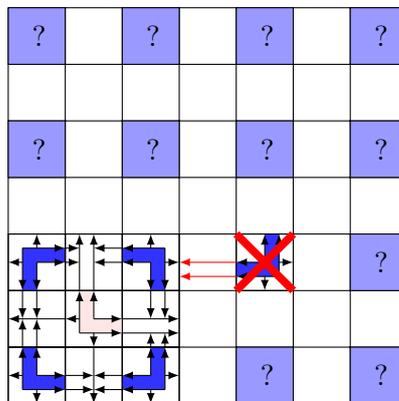


FIGURE 3.2 – Les centres de deux super-tuiles de niveau  $n$  sont espacées d'au moins  $2^n$ . On en déduit que tout élément  $x \in \mathbf{T}_{\mathcal{R}obi}$  n'admet pas de période.

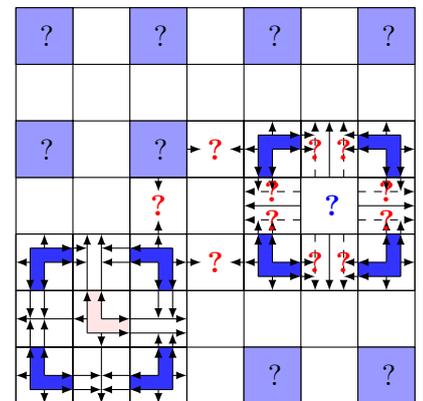
présence d'une super tuile d'ordre  $n$  est forcément située dans un coin d'une super tuile d'ordre  $n + 1$  :



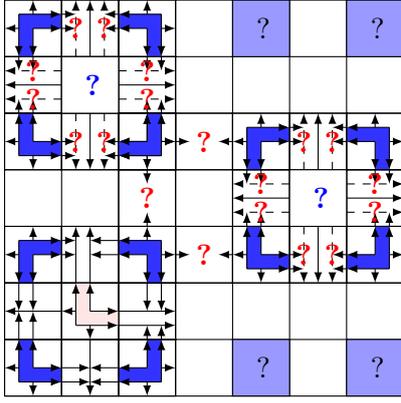
On place une super-tuile d'ordre  $n$  de direction NE.



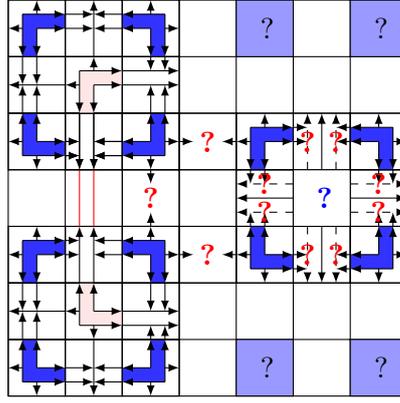
On ne peut pas avoir deux flèches sortantes dans cette direction.



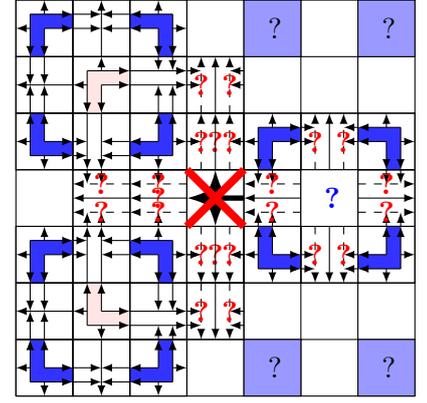
En plaçant un angle dans la direction NE, on obtient forcément une super tuile d'ordre  $n$ .



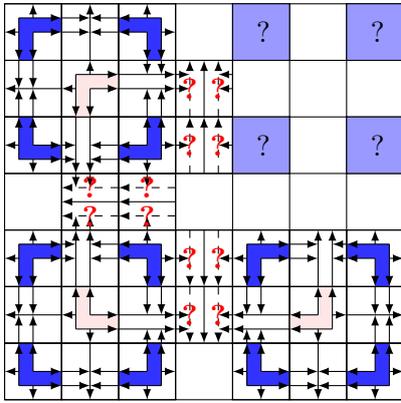
Cela force la présence d'une super-tuile d'ordre  $n$  au dessus de la première.



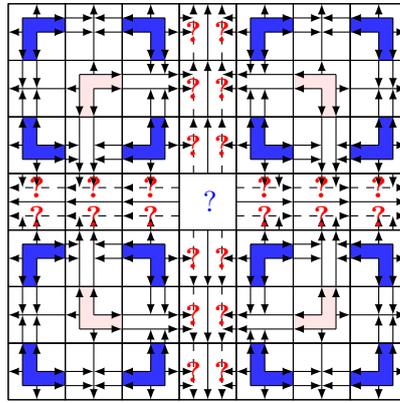
Le type de la super-tuile est forcé par la double flèche de la super tuile sortant de la super-tuile d'ordre  $n$  de direction NE.



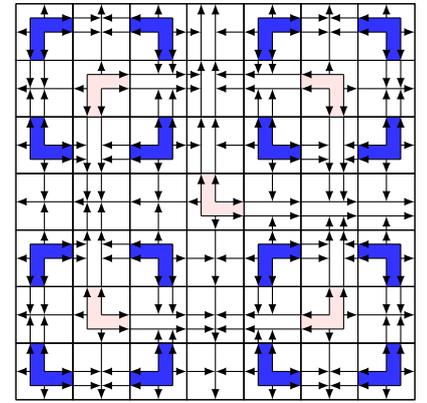
On obtient une incompatibilité sur la tuile centrale.



On a donc une super tuile d'ordre  $n$  de direction NO à droite de la première.



Cela force la dernière super tuile d'ordre  $n$ .



Suivant la direction du coin placé au centre, on obtient une super-tuile d'ordre  $n + 1$  de différente direction.

□

On peut voir une super-tuile d'ordre  $n + 1$  comme un carré dont les angles sont formés par les centres de super-tuiles d'ordre  $n$  qui le forme. La succession des différentes super-tuiles d'ordre  $m$  avec  $m \leq n$  forme une structure hiérarchique qui permet de coder un arbre (voir figure 3.3).

**Proposition 3.1.2.** *Le sous-shift  $\mathbf{T}_{\mathcal{R}obi}$  n'est pas dénombrable.*

*Démonstration.* Lorsque on forme une super tuile d'ordre  $n + 1$  à partir de 4 super-tuile d'ordre  $n$ , on a 4 choix possible pour la cellule centrale. On en déduit que l'on construit un nombre non-dénombrable de configuration (figure 3.4). □

Deux super-tuiles orientées de même ordre mais dans des directions opposées peuvent être collées ensemble mais pas forcément de manière alignée. Par compacité on obtient ce que l'on appelle des *lignes de fracture* d'une configuration  $x \in \mathbf{T}_{\mathcal{R}obi}$  (figure 3.5). Une fracture peut se définir rigoureusement comme l'ensemble des points qui appartiennent à deux tuiles de même ordre mais qui ne sont jamais inclus dans une super-tuile d'ordre supérieur.

**Proposition 3.1.3.**  $\mu(\{x \in \mathbf{T}_{\mathcal{R}obin} \text{ avec une ligne de fracture}\}) = 0$  pour toute mesure de probabilité  $\mu$   $\sigma$ -invariante.

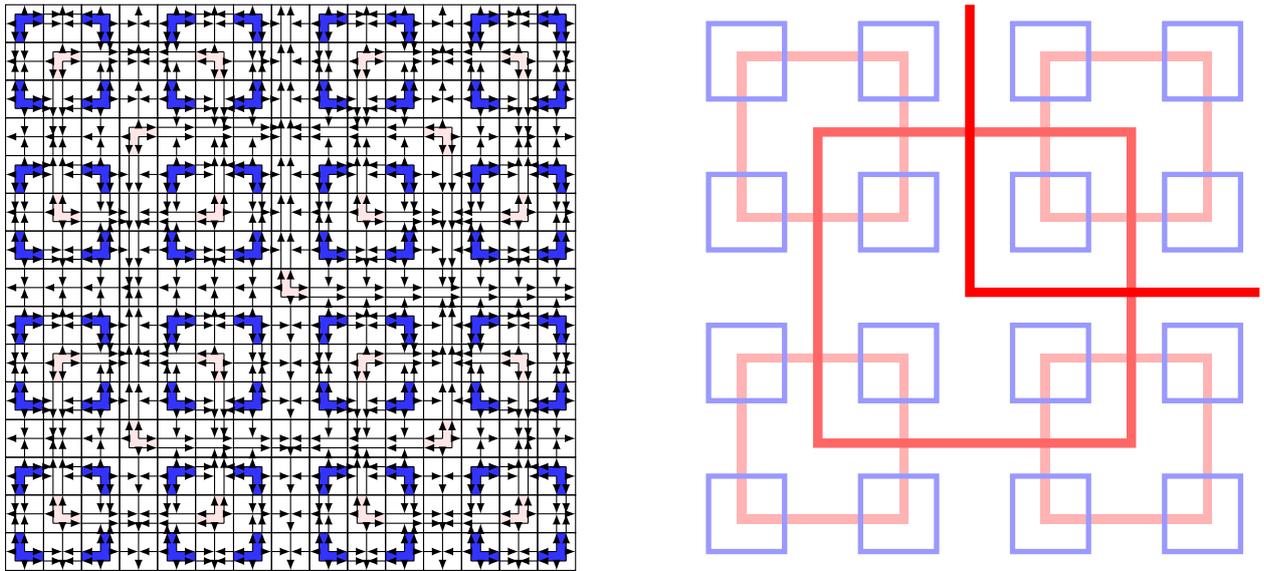
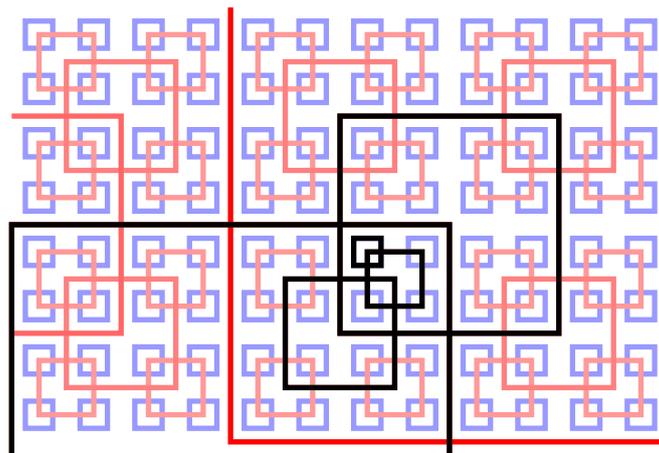


FIGURE 3.3 – Structure hiérarchique d'un pavage de Robinson.



$$\rho(\square) = \begin{matrix} \uparrow\downarrow \\ \leftarrow\rightarrow \end{matrix} \dots$$

FIGURE 3.4 – Liste des choix effectués pour construire le morceau d'une super tuile d'un pavage de Robinson.

*Démonstration.* Par  $\sigma$ -invariance de  $\mu$ , on obtient que tous les ensembles  $\{x \in \mathbf{T}_{Robin} \text{ avec une ligne de fracture avec } i \in \mathbb{Z}\}$  sont de même mesure. Étant donné que  $\mu$  est une mesure de probabilité, on en déduit que la seule valeur possible est 0.  $\square$

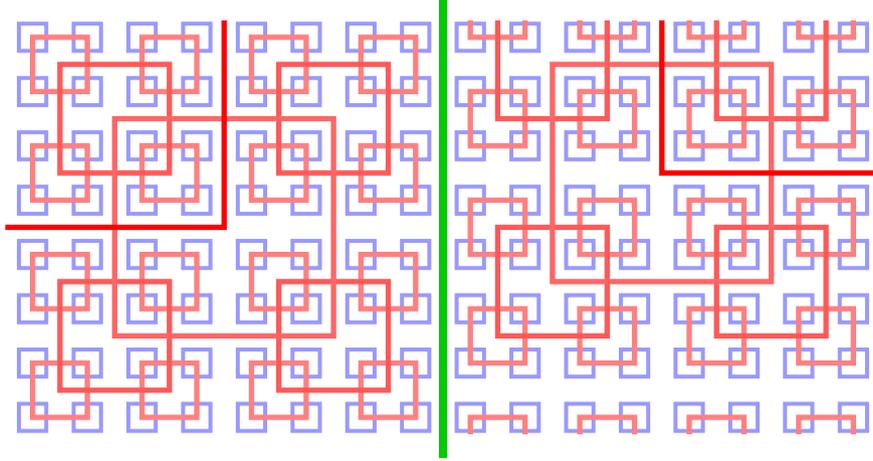


FIGURE 3.5 – Ligne de fracture.

## 3.2 Sous-shift substitutifs

### 3.2.1 Substitutions

Soient  $\mathbf{n} = (\mathbf{n}_1, \dots, \mathbf{n}_d) \in \mathbb{N}^d$  et  $\mathbf{k} = (\mathbf{k}_1, \dots, \mathbf{k}_d) \in \mathbb{N}^d$ , on définit  $\mathbf{n} + \mathbf{k} = (\mathbf{n}_1 + \mathbf{k}_1, \dots, \mathbf{n}_d + \mathbf{k}_d) \in \mathbb{N}^d$ . Étant donné  $\mathbf{k} = (\mathbf{k}_1, \dots, \mathbf{k}_d)$ , on appelle  $\mathbb{U}_{\mathbf{k}}$  le rectangle  $[0; \mathbf{k}_1] \times [0; \mathbf{k}_2] \times \dots \times [0; \mathbf{k}_d]$ .

Sur un alphabet fini  $\mathcal{A}$ , l'ensemble des *motifs rectangulaires* est  $\mathcal{P} = \bigcup_{\mathbf{k} \in \mathbb{N}^d} \mathcal{A}^{\mathbb{U}_{\mathbf{k}}}$ . Une *substitution rectangulaire de dimension  $d$*  est une fonction  $s : \mathcal{A} \rightarrow \mathcal{P}$ . À toute lettre  $a \in \mathcal{A}$ , on associe le vecteur  $\mathbf{k}^s(a) = (\mathbf{k}_1^s(a), \dots, \mathbf{k}_d^s(a))$  tel que  $\text{supp}(s(a)) = \mathbb{U}_{\mathbf{k}^s(a)}$ , ce qui signifie qu'avec notre formalisme, le support du motif  $s(a)$  dépend de la lettre  $a$ . Une substitution multidimensionnelle est *non dégénérée* si  $\mathbf{k}_l^s(a) \geq 1$  pour tout  $l \in [1; d]$  et pour toute lettre  $a \in \mathcal{A}$ .

Soit  $(\mathbf{k}^n)_{n \in \mathbb{Z}}$  une suite de vecteurs  $d$ -dimensionnels. Pour tout  $n \in \mathbb{Z}$  on définit une fonction

$$\phi^{(\mathbf{k}^n)_{n \in \mathbb{Z}}} : \begin{pmatrix} \mathbb{Z}^d & \rightarrow & \mathbb{Z}^d \\ \mathbf{i} & \mapsto & (\phi_1(\mathbf{i}_1), \phi_2(\mathbf{i}_2), \dots, \phi_d(\mathbf{i}_d)) \end{pmatrix}$$

où  $\phi_l(r) = \sum_{j=0}^{r-1} (\mathbf{k}^j)_l$  si  $r \geq 0$  et  $\phi_l(r) = \sum_{j=r}^{-1} (\mathbf{k}^j)_l$  si  $r < 0$ .

Cette fonction  $\phi^{(\mathbf{k}^n)_{n \in \mathbb{Z}}}$  permet de déformer le réseau  $\mathbb{Z}^d$  afin d'en obtenir une partition en rectangles (voir la Figure 3.6).

Soit  $p \in \mathcal{A}^{\mathbb{U}_{\mathbf{k}}}$  un motif rectangulaire de support fini  $\mathbb{U}_{\mathbf{k}} \subset \mathbb{Z}^d$ . On dit que la substitution  $s$  est *compatible* avec le motif  $p$  (respectivement avec la configuration  $x$ ) si pour tous  $\mathbf{i} = (\mathbf{i}_1, \dots, \mathbf{i}_d) \in \mathbb{U}_{\mathbf{k}}$  et  $\mathbf{j} = (\mathbf{j}_1, \dots, \mathbf{j}_d) \in \mathbb{U}_{\mathbf{k}}$  (respectivement  $\mathbf{i} = (\mathbf{i}_1, \dots, \mathbf{i}_d) \in \mathbb{Z}^d$  et  $\mathbf{j} = (\mathbf{j}_1, \dots, \mathbf{j}_d) \in \mathbb{Z}^d$ ) tels que  $\mathbf{i}_l = \mathbf{j}_l$  pour un certain  $l \in [1; d]$ , on a  $\mathbf{k}_l^s(p_{\mathbf{i}}) = \mathbf{k}_l^s(p_{\mathbf{j}})$ . Étant donnée une substitution  $s$  compatible avec une configuration  $x \in \mathbb{Z}^d$ , on peut transformer le réseau  $\mathbb{Z}^d$  en un réseau non régulier grâce à la fonction  $\phi^{(x,s)} = \phi^{(\mathbf{k}^s(x_{(n,\dots,n)}))_{n \in \mathbb{Z}}}$  (voir la Figure 3.7).

Si la substitution  $s$  est compatible avec la configuration  $x$  qui contient un motif  $p$ , la substitution  $s$  agit sur  $p$  et on obtient un motif  $s(p)$  dont le support est

$$\text{supp}(s(p)) = \bigcup_{\mathbf{i} \in \text{supp}(p)} \mathbb{U}_{\mathbf{k}^s(p_{\mathbf{i}})} + \phi^{(x,s)}(\mathbf{i})$$

et tel que

$$\forall \mathbf{i} \in \text{supp}(p), \forall \mathbf{j} \in \text{supp}(s(p_{\mathbf{i}})), \text{ on a } s(p)_{\phi^{(x,s)}(\mathbf{i})+\mathbf{j}} = s(p_{\mathbf{i}})_{\mathbf{j}}.$$

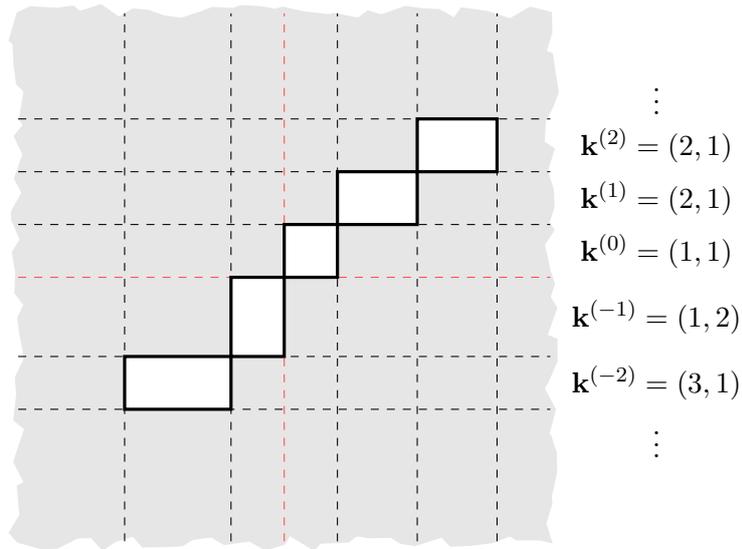


FIGURE 3.6 – Un exemple de suite  $(\mathbf{k}^n)_{n \in \mathbb{Z}}$  de vecteurs 2-dimensionnels qui déforme le réseau  $\mathbb{Z}^2$ .

Ainsi la substitution  $s$  peut facilement s'étendre en une fonction sur les configurations  $s : \left( \begin{array}{c} \mathcal{A}^{\mathbb{Z}^d} \rightarrow \mathcal{A}^{\mathbb{Z}^d} \\ x \mapsto s(x) \end{array} \right)$  où la configuration  $s(x)$  est définie comme expliqué ci-dessus, à condition que la substitution  $s$  soit compatible avec la configuration  $x \in \mathcal{A}^{\mathbb{Z}^d}$ .

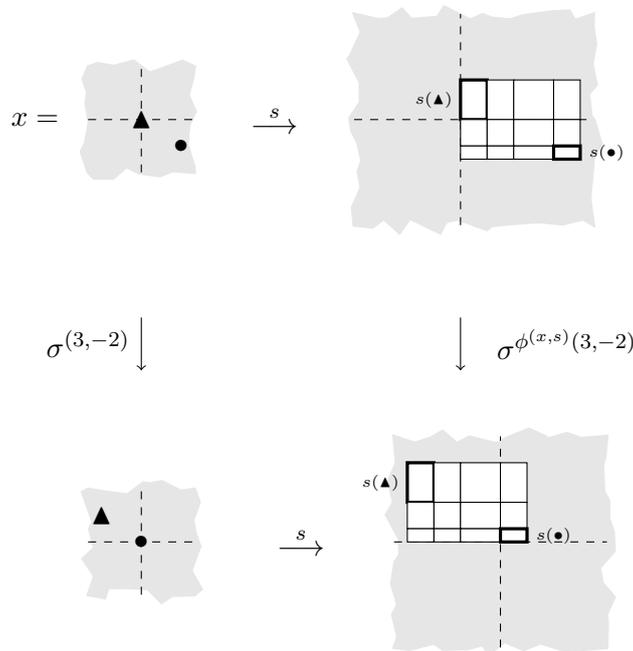


FIGURE 3.7 – Si la substitution  $s$  est compatible avec la configuration  $x$ , alors  $s$  agit sur  $x$  et produit la configuration  $s(x)$ .

*Exemple 3.2.1.* Soient  $\mathcal{A}$  l'alphabet à deux éléments  $\mathcal{A} = \{\circ, \bullet\}$  et  $s$  la substitution données par

les règles suivantes :

$$\circ \mapsto \begin{array}{cc} \circ & \circ \\ \circ & \circ \end{array} \quad \text{et} \quad \bullet \mapsto \begin{array}{cc} \circ & \circ \\ \bullet & \circ \end{array} .$$

Dans cet exemple, le support de  $s(\circ)$  et de  $s(\bullet)$  sont les mêmes, donc  $s$  est compatible avec n'importe quel motif. Par exemple,  $s$  opère sur le motif  $p$  ci-dessous :

$$s : p = \begin{array}{ccc} \circ & \bullet & \bullet \\ \bullet & \circ & \circ \end{array} \mapsto s(p) = \begin{array}{ccc|ccc|cc} \circ & \circ & & \circ & \circ & & \circ & \circ \\ \circ & \circ & & \bullet & \circ & & \bullet & \circ \\ \circ & \circ & & \circ & \circ & & \circ & \circ \\ \bullet & \circ & & \circ & \circ & & \circ & \circ \end{array} .$$

### 3.2.2 Composition de substitutions

Nous venons de voir comment une substitution peut agir sur un motif fini ou sur une configuration infinie. Supposons maintenant qu'il nous est donné un ensemble fini de substitutions, comment le faire agir sur un motif ou une configuration ?

Définissons d'abord la composition de deux substitutions. Soient  $s, s'$  deux substitutions. On dit que  $s'$  est *compatible* avec  $s$  si pour tout motif  $p$  compatible avec  $s$ ,  $s(p)$  est compatible avec  $s'$ . Si  $s'$  est compatible avec  $s$ , on peut alors définir la composition  $s' \circ s$  de la manière suivante : le motif  $s' \circ s(p)$  est obtenu en appliquant la substitution  $s'$  au motif  $s(p)$ . Pour une suite de substitutions  $S_{[0;n]} = (s_0, \dots, s_n)$ , la substitution  $\widehat{S_{[k;n]}}$  est définie par induction de la manière suivante pour tout entier  $k \leq n$  :

$$\begin{cases} \widehat{S_{[k;n]}} = s_n \text{ si } k = n; \\ \widehat{S_{[k;n]}} = s_k \circ \widehat{S_{[k+1;n]}} \text{ si } k < n \text{ et si } s_k \text{ est compatible avec } \widehat{S_{[k+1;n]}}. \end{cases}$$

Notons que si jamais la condition de compatibilité n'est pas vérifiée à une des étapes de la définition de la substitution  $\widehat{S_{[k;n]}}$ , alors celle-ci n'est pas définie.

Considérons désormais  $\mathcal{S}$  un ensemble fini de substitutions de même dimension sur un même alphabet. Dans la suite on présente deux points de vue pour faire agir cet ensemble sur une configuration  $x \in \mathcal{A}^{\mathbb{Z}^d}$ . Dans un premier temps, l'ensemble  $\mathcal{S}$  agit sur une configuration  $x$  au moyen d'une suite de substitutions  $\mathbf{S} = (s_i)_{i \in \mathbb{N}} \in \mathcal{S}^{\mathbb{N}}$ , chaque substitution étant appliquée de manière uniforme à toutes les lettres d'une configuration (section 3.2.3). Mais l'ensemble  $\mathcal{S}$  peut également agir de manière non uniforme sur une configuration  $x$ , la substitution appliquée à une lettre de  $x$  dépendant cette fois de la position de la lettre (section 3.2.4).

### 3.2.3 Sous-shifts $\mathbf{S}$ -adiques

Soit  $\mathcal{S}$  un ensemble fini de substitutions de dimension  $d$  sur le même alphabet  $\mathcal{A}$ , et soit  $\mathbf{S} \in \mathcal{S}^{\mathbb{N}}$  une suite de substitutions. Partant d'une lettre de l'alphabet  $a$ , cette suite de substitutions opère sur  $a$  de la manière suivante. Lors de la  $i^{\text{ième}}$  étape, la substitution  $s_0$  est appliquée à l'ensemble des lettres formant le motif  $s_1 \circ \dots \circ s_i(a)$ . Deux sous-shifts, appelés sous-shifts *S-adiques*, sont définis en se basant sur ce principe.

Dans une première approche combinatoire, on s'intéresse au langage des motifs produits par la suite  $\mathbf{S}$ , appelés les  $\mathbf{S}$ -motifs, que l'on interprète comme les motifs autorisés d'un sous-shift. Le *sous-shift S-adique local* engendré par la suite de substitutions  $\mathbf{S}$ , noté  $\mathbf{T}_{\mathbf{S}}$ , est défini de la façon suivante :

$$\mathbf{T}_{\mathbf{S}} = \left\{ x \in \mathcal{A}^{\mathbb{Z}^d} : \forall p \sqsubset x, \exists a \in \mathcal{A}, \exists n \in \mathbb{N}, \text{ tel que } p \sqsubset \widehat{\mathbf{S}_{[0;n]}}(a) \right\} .$$

C'est donc l'ensemble des configurations telles que tout motif apparaissant dans la configuration apparaît également dans un  $\mathbf{S}$ -motif.

L'autre approche, correspondant à une vision dynamique, consiste à faire agir la suite de substitutions  $\mathbf{S}$  sur l'ensemble des configurations  $\mathcal{A}^{\mathbb{Z}^d}$ . Le *sous-shift  $S$ -adique global* engendré par la suite de substitutions  $\mathbf{S}$ , noté  $\mathbf{T}'_{\mathbf{S}}$ , est alors :

$$\mathbf{T}'_{\mathbf{S}} = \left\{ x \in \mathcal{A}^{\mathbb{Z}^d} : \forall n \in \mathbb{N}, \exists y \in \mathcal{A}^{\mathbb{Z}^d}, \exists \mathbf{i} \in \mathbb{Z}^d \text{ tel que } x = \sigma^{\mathbf{i}} \left( \widehat{\mathbf{S}}_{[0;n]}(y) \right) \right\}.$$

Il s'agit de l'ensemble des configurations pour lesquelles on peut trouver, à une translation près, un antécédent par chacune des itérées de la substitution  $s$ .

Il est aisé de vérifier que si  $\mathbf{S}$  est une suite de substitutions, alors  $\mathbf{T}_{\mathbf{S}} \subseteq \mathbf{T}'_{\mathbf{S}}$ .

*Exemple 3.2.2.* Soit  $s$  la substitution de l'exemple 3.2.1, on considère la suite  $\mathbf{S} = s^{\mathbb{N}}$ . Alors le sous-shift substitutif local associé à cette substitution est :

$$\mathbf{T}_{\{s\}} = \left\{ \circ^{\mathbb{Z}^2} \right\},$$

tandis que son sous-shift substitutif global est :

$$\mathbf{T}'_s = \left\{ \circ^{\mathbb{Z}^2} \right\} \cup \left\{ \sigma^i(x_{\bullet}), i \in \mathbb{Z}^2 \right\},$$

où la configuration  $x_{\bullet}$  est telle que  $x_{(i,j)} = \circ$  si  $i \neq 0$  ou  $j \neq 0$  et  $x_{(0,0)} = \bullet$ . Cette configuration n'appartient pas au sous-shift substitutif local  $\mathbf{T}_{\{s\}}$  car le motif  $\begin{array}{ccc} \circ & \circ & \circ \\ \circ & \bullet & \circ \\ \circ & \circ & \circ \end{array}$  n'apparaît dans aucun des  $s$ -motifs.

Si  $s$  est primitive,  $\mathbf{T}_{\{s\}} = \mathbf{T}'_s$ .

### 3.2.4 Sous-shifts substitutifs non déterministes

Soit  $\mathcal{S}$  un ensemble de substitutions; il est possible de faire agir cet ensemble sur une configuration de manière non déterministe, ou plus précisément de manière non uniforme. Étant donné un motif  $p \in \mathcal{A}^{\mathbb{U}}$ , on applique une substitution sur chacune de ses lettres, qui peut être différente selon la position de la lettre dans le motif.

Si  $\mathbb{U} \subset \mathbb{Z}^d$  est un support fini, on appelle *motif de substitution* un élément  $\mathbf{s} \in \mathcal{S}^{\mathbb{U}}$ . On dit que le motif de substitution  $\mathbf{s} \in \mathcal{S}^{\mathbb{U}}$  est *compatible* avec un motif  $p \in \mathcal{A}^{\mathbb{U}}$  qui apparaît dans  $x \in \mathcal{A}^{\mathbb{Z}^d}$  si pour tous  $\mathbf{i} = (\mathbf{i}_1, \dots, \mathbf{i}_d) \in \mathbb{U}$  et  $\mathbf{j} = (\mathbf{j}_1, \dots, \mathbf{j}_d) \in \mathbb{U}$  tel que  $\mathbf{i}_l = \mathbf{j}_l$  pour un certain  $l \in [1; d]$ , on a  $\mathbf{k} = \mathbf{s}_i^{\mathbf{i}}(p_i) = \mathbf{k}_l^{\mathbf{s}_j}(p_j)$ .

Si le motif de substitution  $\mathbf{s} \in \mathcal{S}^{\mathbb{U}_k}$  est compatible avec un motif  $p \in \mathcal{A}^{\mathbb{U}_k}$ , il agit sur  $p$  et on obtient le motif  $\mathbf{s}(p)$ , dont le support est  $\text{supp}(\mathbf{s}(p)) = \bigcup_{i \in \text{supp}(p)} \mathbb{U}_{\mathbf{k}^{\mathbf{s}_i}(p_i)} + \phi^{(x, \mathbf{s})}(\mathbf{i})$ , défini par :

$$\forall \mathbf{i} \in \text{supp}(p), \forall \mathbf{j} \in \text{supp}(\mathbf{s}_i(p_i)), \quad \mathbf{s}(p)_{\phi^{(x, \mathbf{s})}(\mathbf{i}) + \mathbf{j}} = \mathbf{s}_i(p_i)_{\mathbf{j}}.$$

*Exemple 3.2.3.* Soit  $\mathcal{S} = \{s_1, s_2, s_3, s_4\}$  un ensemble de substitutions de dimension 2 définies sur l'alphabet  $\mathcal{A} = \{\circ, \bullet\}$ , et dont les règles sont les suivantes :

$$\begin{array}{l} s_1 : \circ \mapsto \begin{array}{cc} \circ & \circ \\ \circ & \circ \end{array} \quad \text{et} \quad \bullet \mapsto \begin{array}{cc} \circ & \circ \\ \bullet & \circ \end{array}, \quad s_2 : \circ \mapsto \begin{array}{ccc} \circ & \bullet & \circ \\ \circ & \bullet & \circ \end{array} \quad \text{et} \quad \bullet \mapsto \begin{array}{ccc} \circ & \circ & \circ \\ \bullet & \circ & \circ \end{array} \\ s_3 : \circ \mapsto \begin{array}{cc} \circ & \circ \\ \bullet & \circ \end{array} \quad \text{et} \quad \bullet \mapsto \begin{array}{cc} \circ & \circ \\ \circ & \bullet \end{array}, \quad s_4 : \circ \mapsto \begin{array}{ccc} \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \end{array} \quad \text{et} \quad \bullet \mapsto \begin{array}{ccc} \circ & \circ & \circ \\ \circ & \circ & \circ \\ \bullet & \bullet & \bullet \end{array}. \end{array}$$

Considérons le motif  $p$  dessiné ci-dessous. Alors les motifs de substitution  $\mathbf{s}$  et  $\mathbf{s}'$  sont compatibles avec  $p$  et définissent deux motifs  $\mathbf{s}(p)$  et  $\mathbf{s}'(p)$ , tandis que le motif de substitution  $\mathbf{s}''$  n'est pas compatible avec  $p$ .

$$p = \begin{array}{cccc} \circ & \bullet & \bullet & \bullet \\ \bullet & \bullet & \circ & \circ \end{array}$$

$$\begin{aligned}
\mathbf{s} &= \begin{array}{cccc} s_1 & s_1 & s_2 & s_1 \\ s_3 & s_3 & s_4 & s_3 \end{array}, \mathbf{s}' = \begin{array}{cccc} s_1 & s_1 & s_1 & s_1 \\ s_3 & s_3 & s_3 & s_3 \end{array}, \mathbf{s}'' = \begin{array}{cccc} s_1 & s_1 & s_2 & s_1 \\ s_3 & s_3 & s_4 & s_1 \end{array} \\
\mathbf{s}(p) &= \frac{\begin{array}{ccc|ccc|ccc} \circ & \circ & & \circ & \circ & & \circ & \circ & & \circ & \circ \\ \circ & \circ & & \circ & \circ & & \bullet & \circ & & \bullet & \circ \\ \hline \circ & \circ & & \circ & \circ & & \bullet & \bullet & \bullet & \circ & \circ \\ \circ & \bullet & & \circ & \bullet & & \bullet & \bullet & \bullet & \bullet & \circ \\ \bullet & \bullet & & \bullet & \bullet & & \circ & \circ & \circ & \circ & \bullet \end{array}}{\bullet & \bullet & & \bullet & \bullet & & \circ & \circ & \circ & \circ & \bullet}, \mathbf{s}'(p) = \frac{\begin{array}{ccc|ccc|ccc} \circ & \circ & & \circ & \circ & & \circ & \circ & & \circ & \circ \\ \bullet & \circ & & \circ & \circ & & \circ & \circ & & \circ & \bullet \\ \hline \circ & \bullet & & \bullet & \bullet & & \bullet & \bullet & \bullet & \bullet & \bullet \\ \circ & \circ & & \circ & \circ & & \circ & \bullet & \circ & \circ & \circ \\ \bullet & \circ & & \bullet & \circ & & \circ & \bullet & \circ & \circ & \circ \end{array}}{\bullet & \circ & & \bullet & \circ & & \circ & \bullet & \circ & \circ & \circ}
\end{aligned}$$

L'ensemble des  $\mathcal{S}$ -motifs est défini par induction. Un  $\mathcal{S}$ -motif de niveau 0 est une lettre de l'alphabet  $\mathcal{A}$ , et  $p$  est un  $\mathcal{S}$ -motif de niveau  $n + 1$  s'il existe un  $\mathcal{S}$ -motif  $p' \in \mathcal{A}^{\mathbb{U}}$  de niveau  $n$  et un motif de substitution  $\mathbf{s} \in \mathbb{S}^{\mathbb{U}}$  compatible avec  $p'$  tel que  $\mathbf{s}(p') = p$ . Le support d'un  $\mathcal{S}$ -motif est donc toujours rectangulaire. Les  $\mathcal{S}$ -motifs permettent de définir  $\mathbf{T}_{\mathcal{S}}$ , le *sous-shift substitutif local* engendré par l'ensemble de substitutions  $\mathcal{S}$  :

$$\mathbf{T}_{\mathcal{S}} = \left\{ x \in \mathcal{A}^{\mathbb{Z}^d} : \forall p \sqsubset x, p \text{ est un sous-motif d'un } \mathcal{S}\text{-motif} \right\}.$$

*Remarque.* Pour toute substitution  $s$ , on a  $\mathbf{T}_{\{s\}} = \mathbf{T}_{s^{\mathbb{N}}}$ .

Supposons que  $\mathbf{s} \in \mathcal{S}^{\mathbb{Z}^d}$  est un motif de substitution infini compatible avec une configuration  $x \in \mathcal{A}^{\mathbb{Z}^d}$ . On note  $\mathbf{s}(x)$  la configuration dans  $\mathcal{A}^{\mathbb{Z}^d}$  obtenue en appliquant la substitution  $\mathbf{s}_i$  sur la lettre  $x_i$  pour tout  $i \in \mathbb{Z}^d$ , et en accolant les motifs  $\mathbf{s}_i(x_i)$  ainsi obtenus. On définit  $\mathbf{T}'_{\mathcal{S}}$  le *sous-shift substitutif global* engendré par l'ensemble de substitutions  $\mathcal{S}$  par :

$$\mathbf{T}'_{\mathcal{S}} = \left\{ x \in \mathcal{A}^{\mathbb{Z}^d} : \forall n \in \mathbb{N}, \exists y \in \mathcal{A}^{\mathbb{Z}^d}, \exists (\mathbf{s}_0, \dots, \mathbf{s}_{n-1}) \in (\mathcal{S}^{\mathbb{Z}^d})^n, \exists i \in \mathbb{Z}^d \text{ tel que } x = \sigma^i(\mathbf{s}_0 \circ \dots \circ \mathbf{s}_{n-1}(y)) \right\}.$$

*Remarque.* Les deux inclusions  $\mathbf{T}_{\mathcal{S}} \subseteq \mathbf{T}'_{\mathcal{S}}$  et  $\mathbf{T}'_{\mathcal{S}} \subseteq \mathbf{T}'_{\mathcal{S}}$  sont vérifiées pour toute suite de substitutions  $\mathbf{S}$  sur un ensemble de substitutions  $\mathcal{S}$ .

### 3.2.5 Sous-shift substitutifs et apériodicité

Une substitution  $s$  est à *dérivation unique* si pour tout  $x \in \mathbf{T}'_{\{s\}}$  il existe un unique  $y \in \mathbf{T}'_{\{s\}}$  et un unique  $n \in \mathbb{Z}^d$  tel que  $x = \sigma^n(s(y))$ . Dans ce cas, si une configuration  $x \in \mathbf{T}'_{\{s\}}$  est périodique la période peut être réduite en prenant la pré-image. On en déduit la proposition suivante.

**Proposition 3.2.1.** *Si  $s$  est à dérivation unique alors toute configuration  $x \in \mathbf{T}'_{\{s\}}$  est apériodique.*

### 3.2.6 Substitutions non-déterministes et théorème de Mozes

On dit qu'un ensemble de substitutions  $\mathcal{S}$  est *de type A*, ou *possède la propriété A*, s'il vérifie la condition définie ci-après. Soient  $p$  un  $\mathcal{S}$ -motif et  $l$  un sous-motif de taille  $2 \times 2$  apparaissant dans  $p$ . Supposons qu'il existe une suite de motifs de substitution  $\mathbf{s}_1, \dots, \mathbf{s}_n$  compatible avec le motif  $l$ , qui produit une suite de motifs  $l_0 = l, l_1 = \mathbf{s}_1(l_0), \dots, l_n = \mathbf{s}_n(l_{n-1})$ . Alors il est possible de trouver une suite de motifs de substitution  $\mathbf{s}'_1, \dots, \mathbf{s}'_n$  compatible avec le motif  $p$  telle que les motifs dérivant de  $l$  dans  $p_0 = p, p_1 = \mathbf{s}'_1(p_0), \dots, p_n = \mathbf{s}'_n(p_{n-1})$  sont exactement les  $l_0, l_1, \dots, l_n$  (voir la Figure 3.8).

Cette propriété *A* n'est a priori pas triviale. Il est en effet possible que la suite de motifs de substitution choisie pour  $l$  ne soit pas compatible avec le motif  $p$ . Dans ce cas il est quand même possible de trouver une autre suite de motifs de substitution compatible avec  $p$  et telle que les motifs dérivant de  $l$  sont bien ceux recherchés.

La propriété *A* est tout de même vérifiée par bon nombre d'ensembles de substitutions. Par exemple tout ensemble de substitutions  $\mathcal{S}$  dans lequel l'image d'une lettre par une substitution  $s \in \mathcal{S}$  ne dépend que de la substitution  $s$  possède la propriété *A*. De plus, si l'ensemble  $\mathcal{S}$  est réduit à une unique substitution déterministe, alors  $\mathcal{S}$  possède la propriété *A*.

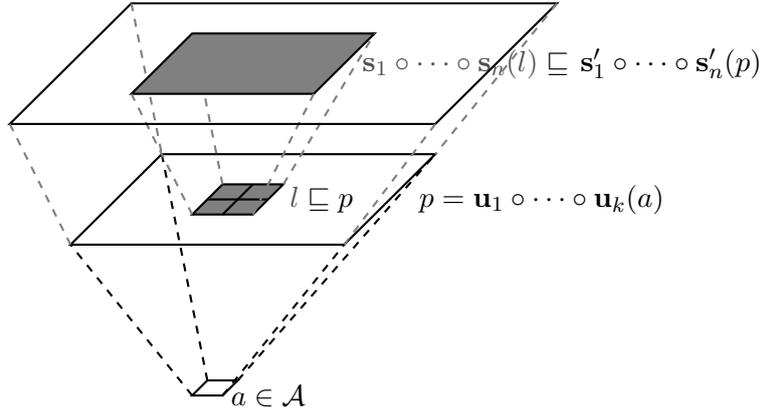


FIGURE 3.8 – Un ensemble de substitutions  $\mathcal{S}$  qui possède la propriété  $A$ .

**Théorème 3.2.2** ([Moz89]). *Soit  $\mathcal{S}$  un ensemble de substitutions multidimensionnelles non dégénérées possédant la propriété  $A$ . Alors le sous-shift  $\mathbf{T}_{\mathcal{S}}$  est sofique.*

*Plus précisément, si  $\mathbf{T}_{\mathcal{S}} \subset \mathcal{A}^{\mathbb{Z}^d}$  il existe un sous shift de type fini  $\mathbf{T}(\mathcal{B}, d, \mathcal{F})$  et un facteur lettre à lettre  $\pi : \mathcal{B} \rightarrow \mathcal{A}$  tels que*

$$\mathbf{T}_{\mathcal{S}} = \mathbf{T}(\mathcal{B}, d, \mathcal{F}).$$

*De plus  $\pi$  est une conjugaison  $\mu$ -presque partout pour toute mesure  $\mu$   $\sigma$ -invariante. On en déduit que  $h(\mathbf{T}(\mathcal{B}, d, \mathcal{F})) = h(\mathbf{T}_{\mathcal{S}})$ .*

*De plus il existe un ensemble de substitutions  $\tilde{\mathcal{S}}$  sur l'alphabet  $\mathcal{B}$  tel que  $\mathbf{T}_{\tilde{\mathcal{S}}} = \mathbf{T}(\mathcal{B}, d, \mathcal{F})$ .*

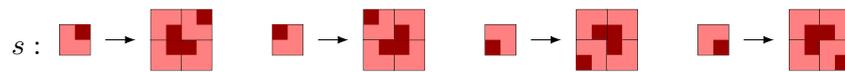
**Corollaire 3.2.3.** *Le théorème 3.2.2 reste valable si on considère le sous-shift  $\mathbf{T}'_{\mathcal{S}}$  à la place de  $\mathbf{T}_{\mathcal{S}}$ . Dans ce cas là on n'a plus besoin de la propriété  $A$ .*

*Idée de la démonstration :* On construit un sous-shift de type fini  $\mathbf{T}(\mathcal{B}, d, \mathcal{F})$  formé de plusieurs couches tel que  $\mathbf{T}_{\mathcal{S}}$  est un facteur de  $\mathbf{T}_{\text{Simul}}$  lorsqu'on conserve l'une d'elles. En voici une description succincte reprise dans l'exemple 3.2.4 :

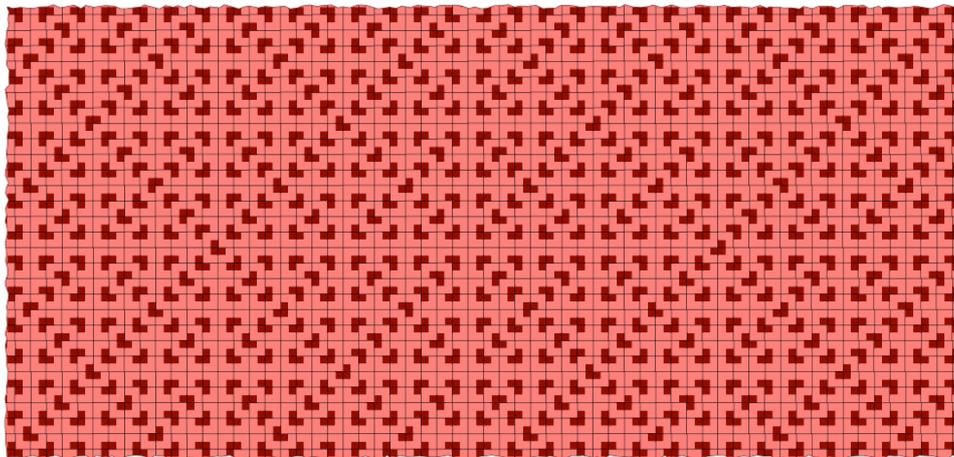
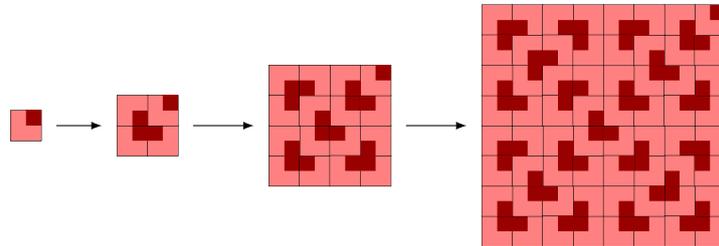
1. Une couche correspondant à l'alphabet  $\mathcal{A}$  qui sert à récupérer le sous-shift  $\mathbf{T}_{\mathcal{S}}$  à l'aide du facteur lettre à lettre.
2. Une couche qui force une structure hiérarchique avec des carrés (ou des rectangles si les substitutions de  $\mathbf{T}_{\mathcal{S}}$  sont rectangulaires) emboîtés similaire au pavage de Robinson.
3. Une couche qui place une tuile de  $\mathcal{A}$  au centre de chaque carré et force les sommets du carré de niveau inférieur associé en fonction de la règle de substitution. Si on est dans un carré d'ordre  $n$ , cette couche permet de dé-substituer  $n$  fois le motif associé.
4. Pour finir, une couche qui relie les carrés de différents ordres de telle sorte que tout les carrés soient bien alignés afin de synchroniser les super-tuiles qui sont de part et d'autre des lignes de fractures.

Si on ne rajoute pas d'autres conditions, étant donné un motif au niveau des lignes de fracture, on pourra le dé-substituer jusqu'à obtenir un motif  $1 \times 1$ ,  $2 \times 1$ ,  $1 \times 2$  ou  $2 \times 2$ . Ainsi le motif initial est un motif de  $\mathbf{T}'_{\mathcal{S}}$ . Si on souhaite avoir un motif de  $\mathbf{T}_{\mathcal{S}}$ , il faut pouvoir dé-substituer une fois de plus et donc vérifier sur la quatrième couche que quatre carrés adjacents sont dans le langage de  $\mathbf{T}_{\mathcal{S}}$ .

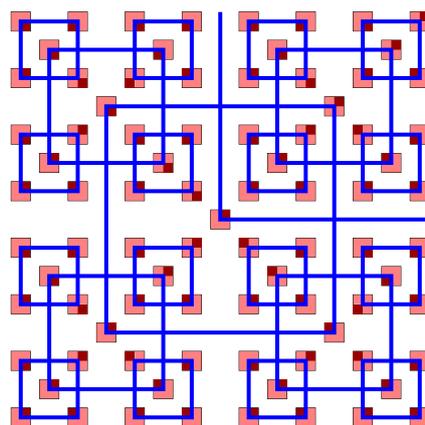
Exemple 3.2.4. Soit  $\mathcal{A} = \{\text{■}, \text{■}, \text{■}, \text{■}\}$ . On considère la substitution suivante :



Ce qui donne en itérant :



A l'aide de la deuxième couche de  $\mathbf{T}_{\text{simul}}$ , on force une structure hiérarchique en plaçant une tuile de  $\mathcal{A}$  au sommet de chaque carré d'ordre  $n + 1$ . Cette tuile est au centre d'un carré d'ordre  $n$ . Puis on force, à l'aide de signaux situés sur les cotés des carrés, que les tuiles de  $\mathcal{A}$  placées sur les sommets du carré d'ordre  $n$  corresponde à la substitution de la tuile placée au centre. Ainsi les tuiles placées à l'ordre 0 peuvent être dé-substitué une infinité de fois :



Pour éviter le problème lié aux lignes de fractures, il faut synchroniser les super tuiles de même ordre adjacentes. Pour cela, on synchronise les carrés de même ordre (cf figure 3.9). Dans ce cas on obtient  $\mathbf{T}'_s$ , pour obtenir  $\mathbf{T}_s$ , il faut de plus forcer les super tuiles adjacente à être dans le langage de  $\mathbf{T}_s$ .

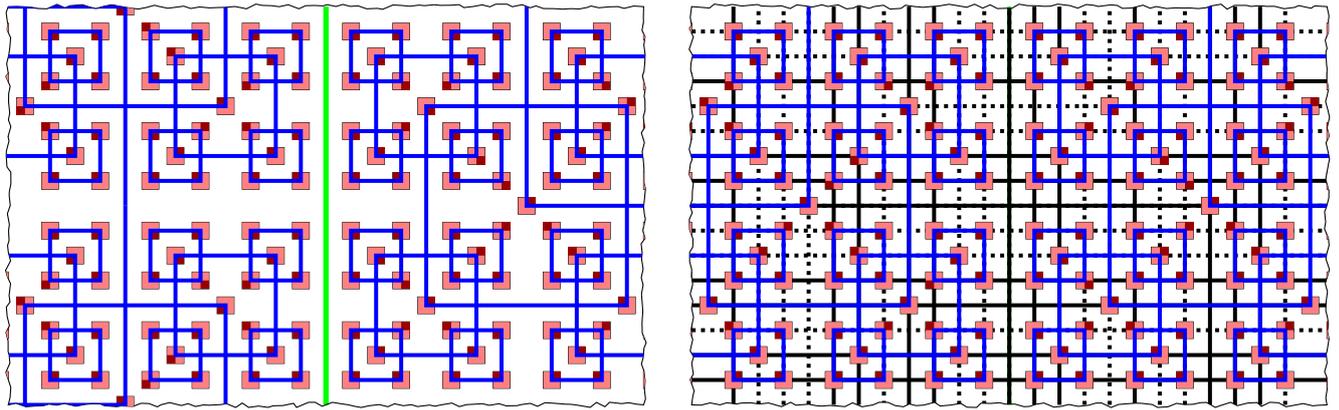


FIGURE 3.9 – A gauche, deux super-tuiles d'ordre 3 ne sont pas face à face, on ne pourra donc pas dé-substituer plus de trois fois. Pour éviter ce problème, on synchronise à droite horizontalement et verticalement les carrés de même ordre.

### 3.3 Pavages substitutifs avec des polygones

De manière analogue, on peut définir une substitution sur des polygones. Étant donnée une famille de polygone  $\mathcal{P}$ , le principe consiste à dilater les polygones puis à les découper de telle sorte qu'ils soient reformés par les polygones de  $\mathcal{P}$ .

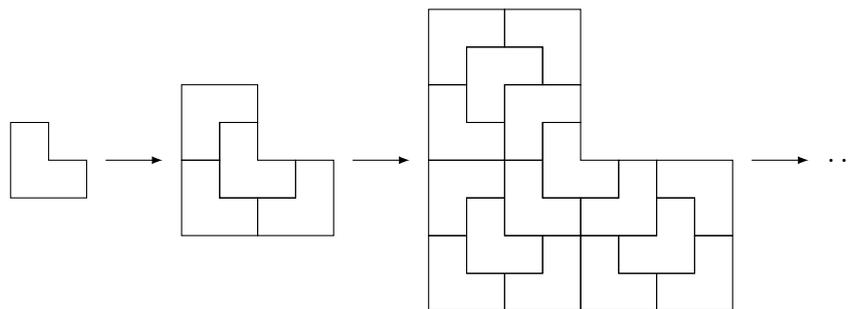


FIGURE 3.10 – Exemple de substitution géométrique.

**Théorème 3.3.1** ([GS98]). *L'espace des pavages défini par une substitution de polygones est sofique.*

Ce théorème peut se généraliser en utilisant les mêmes méthodes pour des substitutions définis de manière combinatoire [FO10] ou à l'aide de graphe [FBS].

### 3.4 Pavages apériodique à 13 tuiles de Kari-Culik

Les pavages apériodiques connus jusqu'en 1996 étaient basés sur une structure hiérarchique et une substitution pouvait engendrer un pavage conjugué. Si on se ramène à un jeu de tuile de Wang, généralement le nombre de tuiles nécessaire pour réaliser un pavage apériodique est important. J. Kari [Kar96] exhiba un pavage apériodique à partir de 14 tuiles de Wang et par cette méthode K. Culik [II96] a obtenu un jeu de 13 tuiles pavant uniquement de manière apériodique.

### 3.4.1 Tuiles de multiplication

Une tuile  $\begin{bmatrix} n \\ w & e \\ s \end{bmatrix}$  est une *tuile de multiplication par  $\lambda$*  si  $\lambda s + w = n + e$ .

Pour  $x \in \mathbb{R}$ , on note  $B(x)_i = \lfloor ix \rfloor - \lfloor (i-1)x \rfloor$  pour  $i \in \mathbb{Z}$  la *représentation équilibrée de  $x$* . On remarque que  $B(x)_i \in \{\lfloor x \rfloor, \lfloor x \rfloor + 1\}$ , de plus on montre que  $x = \lim_{n \rightarrow \infty} \frac{1}{2n+1} \sum_{i=-n}^n B(x)_i$ .

Un transducteur est un quadruplet  $\mathbf{Trans} = (Q, \Gamma, \Gamma', \Delta)$  où  $Q$  est l'ensemble d'état,  $\Gamma$  l'alphabet d'entrée,  $\Gamma'$  l'alphabet de sortie et  $\Delta \subset Q \times \Gamma \times \Gamma' \times \Delta$  est l'ensemble de transitions. Un transducteur  $\mathbf{Trans}$  peut être représenté par un graphe orienté étiqueté avec pour sommet les éléments de  $Q$  et avec une arête de l'état  $q$  vers l'état  $q'$  étiqueté par  $(a/b)$  si  $(q, a, b, q') \in \Delta$ . Le transducteur  $\mathbf{Trans}$  transforme un élément de  $x \in \Gamma^{\mathbb{Z}}$  en un élément  $\mathbf{Trans}(x) \in \Gamma'^{\mathbb{Z}}$  s'il existe  $s \in Q^{\mathbb{Z}}$  tel que pour tout  $i \in \mathbb{Z}$  on ait  $(s_{i-1}, x_i, \mathbf{Trans}(x)_i, s_i) \in \Delta$ .

Pour un rationnel positif  $\lambda = \frac{n}{m}$ , on construit le transducteur  $\mathbf{Trans}_\lambda$  qui multiplie par  $\lambda$  un réel  $x$  écrit en représentation équilibrée  $B(x)$  afin d'obtenir  $B(\lambda x)$ . On pose  $\Gamma = \{\lfloor x \rfloor, \lfloor x \rfloor + 1\}$  et  $\Gamma' = \{\lfloor \lambda x \rfloor, \lfloor \lambda x \rfloor + 1\}$  correspondant respectivement à l'alphabet d'entrée et de sortie de  $\mathbf{Trans}_\lambda$ . Les états de  $\mathbf{Trans}_\lambda$  sont  $Q = \{\lambda \lfloor x \rfloor - \lfloor \lambda x \rfloor : x \in \mathbb{R}\} = \{-\frac{n-1}{m}, -\frac{n-2}{m}, \dots, \frac{n-2}{m}, \frac{n-1}{m}\}$  et l'on effectue une transition de  $q \in Q$  avec l'entrée  $a$  et la sortie  $b$  vers  $q'$  si et seulement si  $q' = q + \lambda a - b$ .

Après avoir lu les entrées  $\dots B(x)_{i-2} B(x)_{i-1}$  et écrit les sorties  $\dots B(\lambda x)_{i-2} B(\lambda x)_{i-1}$ , on suppose que  $\mathbf{Trans}_\lambda$  est dans l'état  $q_{i-1} = \lambda \lfloor (i-1)x \rfloor - \lfloor \lambda(i-1)x \rfloor \in Q$ . Sur l'entrée  $B(x)_i$ , la machine sort  $B(\lambda x)_i$  et se retrouve dans l'état

$$\begin{aligned} q_{i-1} + \lambda B(x)_i - B(\lambda x)_i &= \lambda \lfloor (i-1)x \rfloor - \lfloor \lambda(i-1)x \rfloor + \lambda \lfloor ix \rfloor - \lambda \lfloor (i-1)x \rfloor - \lfloor \lambda ix \rfloor + \lfloor \lambda(i-1)x \rfloor \\ &= \lambda \lfloor ix \rfloor - \lfloor \lambda ix \rfloor \\ &= q_i \in Q \end{aligned}$$

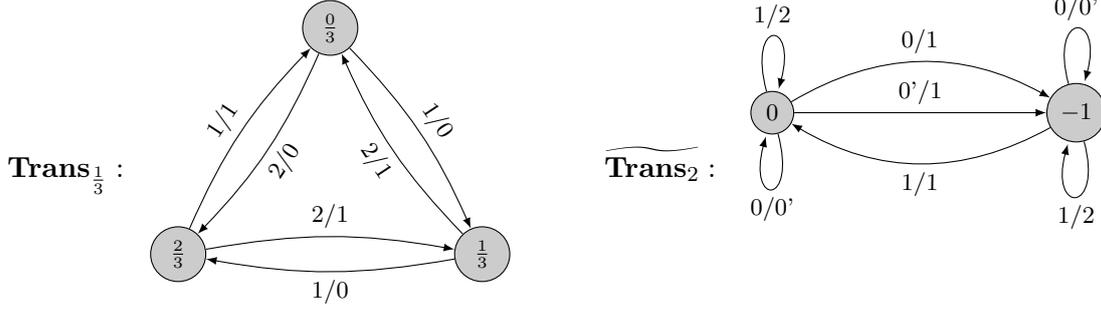
Ainsi la machine transforme bien la suite  $B(x)$  en  $B(\lambda x)$ .

Étant donné un transducteur  $\mathbf{Trans}_\lambda$ , pour toute transition  $(q, a, b, q') \in \Delta$  on définit la tuile  $t(q, a, b, q') = \begin{bmatrix} b \\ q & q' \\ a \end{bmatrix}$ . Par définition, on obtient une tuile de multiplication par  $\lambda$ .

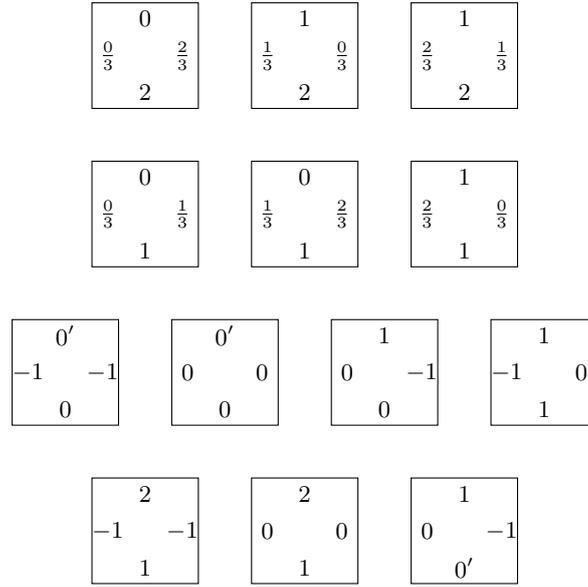
### 3.4.2 Pavage à 13 tuiles

On considère le transducteur  $\mathbf{Trans}_{\frac{1}{3}}$  construit pour multiplier par  $\frac{1}{3}$  en utilisant  $\{1, 2\}$  comme alphabet d'entrée et  $\{0, 1\}$  comme alphabet de sortie. Cela signifie que  $B(x)$  et  $B(\frac{1}{3}x)$  sont en relation par  $\mathbf{Trans}_{\frac{1}{3}}$  pour tout réel vérifiant  $1 \leq x \leq 2$  et  $0 \leq \frac{1}{3}x \leq 1$  c'est à dire  $x \in [1, 2]$ . De même on considère le transducteur  $\mathbf{Trans}_2$  construit pour multiplier par 2 en utilisant  $\{0, 1\}$  comme alphabet d'entrée et  $\{0, 1, 2\}$  comme alphabet de sortie. Cela signifie que  $B(x)$  et  $B(2x)$  sont en relation par  $\mathbf{Trans}_2$  pour tout réel  $x \in [0, 1]$ .

Le but est d'itérer  $\mathbf{Trans}_2$  et  $\mathbf{Trans}_{\frac{1}{3}}$  sans utiliser plus de deux fois consécutives  $\mathbf{Trans}_2$  afin de simuler la fonction affine définie par  $f(x) = \begin{cases} 2x & \text{si } x \in [\frac{1}{3}, 1[ \\ \frac{1}{3}x & \text{si } x \in [1, 2] \end{cases}$ . Pour cela, on introduit un nouveau symbole d'entrée/sortie noté  $0'$  pour définir le transducteur  $\widetilde{\mathbf{Trans}}_2$ . On a alors les deux transducteurs suivants :



A chaque transition de ces transducteurs on obtient une tuile, cela forme les 13 tuiles du pavage Kari-Culik, notés  $\mathcal{KC}$  :



On remarque que les six tuiles du haut de  $\mathcal{KC}$  sont de type  $\lambda = \frac{1}{3}$  et les sept du dessous de type  $\lambda = 2$ . On note  $\mathbf{T}_{\mathcal{KC}} \subset \mathcal{KC}^{\mathbb{Z}^2}$  le SFT engendré par les tuiles de Wang précédentes.

**Théorème 3.4.1.** *Le sous-shift de type fini  $\mathbf{T}_{\mathcal{KC}}$  est non vide et toute configuration  $x \in \mathbf{T}_{\mathcal{KC}}$  est apériodique.*

*Démonstration.* On montre les deux choses suivantes :

- $\mathbf{T}_{\mathcal{KC}} \neq \emptyset$  : Pour toute entrée  $B(x)$  avec  $x \in [\frac{1}{3}, 2]$ , on a un transducteur qui calcule  $B(2x)$  si  $x \in [\frac{1}{3}, 1]$  et  $B(\frac{1}{3}x)$  si  $x \in [1, 2]$ . Dans les deux cas on peut appliquer autant de fois qu'on veut pour obtenir des motifs arbitrairement grand.

- Toute configuration de  $\mathbf{T}_{\mathcal{KC}}$  est apériodique : On remarque que chaque ligne est soit du type  $\lambda = \frac{1}{3}$  soit du type  $\lambda = 2$  et toute configuration  $x \in \mathbf{T}_{\mathcal{KC}}$  contient les deux types.

Si  $x \in \mathbf{T}_{\mathcal{KC}}$  est une configuration périodique de période  $u \times v$ . Comme  $\lambda_v n(x_{i,v}) + e((x_{i,v})) = s(x_{i,v}) + w(x_{i,v})$  pour tout  $i \in [1, u]$  et  $w(1, v) = e(u, v)$ , on a

$$\lambda_v \sum_{i=1}^u n(x_{i,v}) = \sum_{i=1}^u s(x_{i,v}) = \sum_{i=1}^u n(x_{i,v-1})$$

et par récurrence

$$\lambda_1 \dots \lambda_v \sum_{i=1}^u n(x_{i,v}) = \sum_{i=1}^u s(x_{i,1}) = \sum_{j=1}^u n(x_{i,0}) = \sum_{i=1}^u n(x_{i,v})$$

On a donc  $1 = \lambda_1 \dots \lambda_v = \frac{2^k}{3^{u-k}}$  avec  $k \in [1, v - 1]$ , on en déduit une contradiction. □

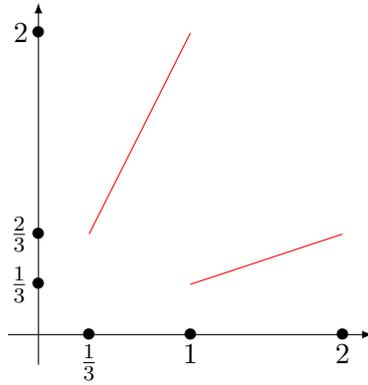


FIGURE 3.11 – Graphe de la fonction  $f$ .

## Chapitre 4

# Pavages et modèle de calcul

### 4.1 Problèmes de décision

Une question apparue naturellement est de savoir si étant donné un ensemble fini de motif  $\mathcal{F}$ , est ce que  $\mathbf{T}(\mathcal{A}, d, \mathcal{F}) = \emptyset$ ? Ce problème s'intitule le problème de décision du domino :

“Est-il possible de déterminer si un jeu fini de tuile de Wang pave le plan?” [Wan61]

Un *problème de décision* est un problème qui prend en entrée une *instance* du problème et répond en sortie “oui” ou “non” suivant si l'instance répond au problème ou pas. L'entrée doit être codée de manière à être traitée informatiquement, généralement une suite finie de bits (ou de caractères pris dans un alphabet fini). On cherche à savoir si une instance du problème peut être résolue par un *programme*, c'est à dire une suite finie et non-ambiguë d'opérations ou d'instructions. Un programme peut être codé informatiquement par une suite finie de bits (ou de caractères pris dans un alphabet fini). Il existe différentes formalisations nous utiliserons par la suite le formalisme des machines de Turing.

Un problème de décision est *décidable* s'il existe un programme permettant de le résoudre sur n'importe quelle entrée et *indécidable* sinon.

On note que si un problème n'a qu'un nombre fini d'instances alors il est forcément décidable car il existe un programme qui à chaque instance attribue une réponse. Par exemple le problème “Est-ce que la conjecture de Riemann est vrai” est décidable. C'est soit le programme qui répond toujours “oui” soit le programme qui répond toujours “non”. De même le problème “Est-il possible de déterminer si un jeu fini de tuile de Wang comportant au plus  $M$  couleurs pave le plan?” est décidable.

Un problème de décision est *semi-décidable* s'il existe un programme qui répond “oui” pour une instance positive du problème et tourne indéfiniment sinon. Ainsi il n'est pas possible de savoir si une instance est négative car on ne sait pas si le programme ne va pas s'arrêter plus tard.

**Proposition 4.1.1.** *Le complémentaire du problème de domino “Est-ce qu'un jeu de tuile de Wang ne pave pas le plan” est semi-décidable.*

*Démonstration.* On énumère les entiers. Pour chaque entier  $n$ , on énumère tous les motifs de taille  $n \times n$ . Si tous ces motifs admettent un motif interdit alors le jeu de tuile ne pave pas le plan, sinon on passe au pavé suivant.  $\square$

**Proposition 4.1.2.** *Le problème du domino est décidable en dimension 1.*

*Démonstration.* On écrit le graphe du langage d'ordre correspondant à la taille du plus grand motif interdit. Si le graphe admet une boucle alors le SFT est non vide.  $\square$

**Proposition 4.1.3.** *Un problème de décision est décidable si et seulement si ce problème et son complémentaire sont semi-décidables.*

*Démonstration.* On fait tourner en parallèle les deux programmes issus de la semi-décidabilité et le calcul s'arrête lorsque l'un des deux s'arrête on peut alors répondre à la question.  $\square$

**Théorème 4.1.4** (Turing 1936). *Le problème de décision "Est-ce que un programme s'arrête sur une entrée donnée?", appelé problème de l'arrêt est semi-décidable mais il n'est pas décidable.*

*Démonstration.* Ce problème est clairement semi-décidable, il suffit de lancer le programme à tester sur l'entrée souhaitée et attendre de voir s'il s'arrête.

Supposons qu'il existe un programme **SuperProg** répondant au problème de l'arrêt. C'est à dire :

$$\text{SuperProg}(P, w) = \begin{cases} \text{"oui"} & \text{si } P(w) \text{ s'arrête} \\ \text{"non"} & \text{si } P(w) \text{ ne s'arrête pas} \end{cases}$$

On considère maintenant le programme **Bug** qui prend en entrée un programme  $P$  et répond :

$$\text{Bug}(P) = \begin{cases} \text{"oui"} & \text{si } \text{SuperProg}(P, P) = \text{"non"} \\ \text{boucle indéfiniment} & \text{sinon} \end{cases}$$

On obtient un paradoxe en faisant tourner **Bug** sur lui même, en effet  $\text{Bug}(\text{Bug})$  s'arrête si et seulement si il boucle indéfiniment et donc ne s'arrête pas.  $\square$

Pour montrer qu'un problème  $P$  est indécidable, on se ramène généralement à un problème indécidable connu noté  $Q$ , par exemple le problème de l'arrêt. Pour cela on suppose que  $P$  est décidable et on construit un programme qui résout  $Q$  en utilisant comme sous-routine le programme qui résout  $P$ .

**Corollaire 4.1.5** (Turing 1936). *Le problème de décision "Est-ce que un programme s'arrête sur l'entrée vide?" est semi-décidable mais il n'est pas décidable.*

*Démonstration.* Supposons qu'un programme  $P_0$  décide si un programme s'arrête sur l'entrée vide. Étant donnée un programme  $P$  et une entrée  $w$ . On considère alors le programme **SuperProg** qui étant donnée l'entrée  $(P, w)$  place en mémoire  $w$  puis simule  $P$  sur l'entrée  $w$ . On en déduit **SuperProg** qui décide le problème de l'arrêt ce qui est impossible.  $\square$

Il y a un parallèle étroit avec les théorèmes d'incomplétude de Gödel (correspondance preuve/programme de Curry-Howard, 1958/69), datant des mêmes années (1931) et reposant aussi sur l'auto-référencement. On peut montrer que non seulement l'arrêt mais toute propriété non triviale est indécidable de Rice, 1953).

## 4.2 Machines de Turing

Une machine de Turing est un modèle de calcul simple. Elle est constituée d'un *ruban* indexé par  $\mathbb{Z}$  qui contient en chaque site un élément d'un alphabet fini  $\Gamma$ . Cela peut donc être vu comme une configuration  $x \in \Gamma^{\mathbb{Z}}$  qui constitue la mémoire de notre système de calcul. Un élément  $\# \in \Gamma$  appelé *symbole blanc* correspond à l'absence d'information. Sur la configuration initiale il y a seulement l'entrée, codée à l'aide de  $\Gamma$  et le symbole  $\#$ .

A chaque unité de temps, l'*unité de contrôle* a accès à un site  $i \in \mathbb{Z}$ . Suivant la valeur de ce site  $x_i$  et l'*état*  $q \in Q$ , où  $Q$  est un nombre fini d'états, dans lequel l'unité de contrôle se trouve, la machine modifie le symbole sur le ruban en position  $i$  (elle peut re-écrire éventuellement le

même), change l'état où elle se trouve, puis déplace l'unité de contrôle sur la cellule  $i - 1$ ,  $i$  ou  $i + 1$ . Toutes les actions possibles sont spécifiées par la *fonction de transition*  $\delta : \Gamma \times Q \rightarrow \Gamma \times Q \times \{\leftarrow, \uparrow, \rightarrow\}$ . L'ensemble des états  $Q$  contient l'état initial  $q_0$  et l'état final  $q_F$ .

Ainsi une machine de Turing  $\mathcal{M}$  est la donnée de  $(\Gamma, \#, Q, q_0, q_F, \delta)$ . Une *configuration* de  $\mathcal{M}$  sera un élément de  $Q \times \mathbb{Z} \times \Gamma^{\mathbb{Z}}$  qui spécifie l'état où se trouve la machine, la position de la tête de lecture ainsi que l'état du ruban. L'état initial étant  $(q_0, 0, x)$  où  $x$  contient l'entrée écrit sur une partie finie du ruban puis le symbole  $\#$ . La fonction de transition  $\delta$  modifie l'état d'une machine pour le transformé en un autre ou le ruban a été modifié seulement à  $n$  endroits. La machine s'arrête si l'état  $q_F$  est atteint.

Il existe plusieurs variantes de modèles de machine de Turing qui ont tous la même puissance de calcul. On peut par exemple supposer que le ruban est infini seulement d'un côté, une configuration de  $\mathcal{M}$  sera alors un élément de  $\Gamma^{\mathbb{N}}$ .

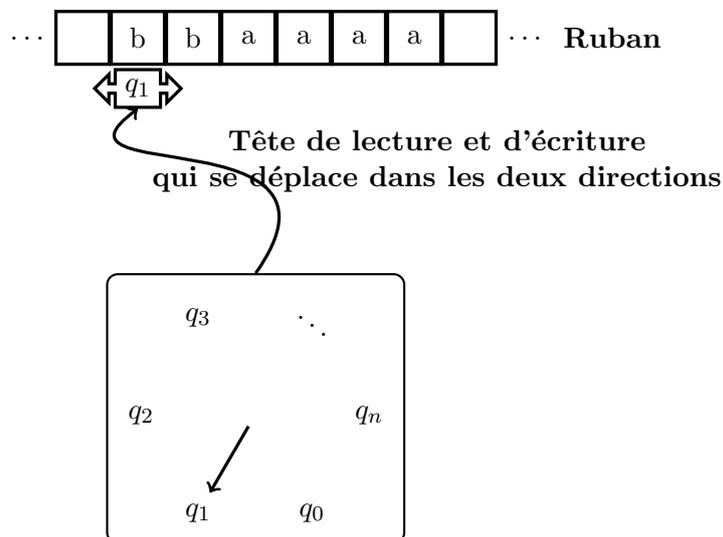


FIGURE 4.1 – Modèle de machine de Turing.

*Exemple 4.2.1.* Considérons la machine de Turing  $\mathcal{M}_{\text{ex}}$  qui énumère sur son ruban les mots du langage  $\mathcal{L} = \{a^n b^n : n \in \mathbb{N}\}$  séparés par le symbole  $\|$  et ne s'arrête jamais. Cette machine travaille sur l'alphabet à trois lettres  $\Gamma = \{a, b, \|\}$  et la tête de calcul peut être dans cinq états  $Q = \{q_0, q_{a+}, q_{b+}, q_{b++}, q_{\|\}$ . Lorsque la tête de calcul est dans l'état  $q_{a+}$ , elle va ajouter un symbole au mot déjà inscrit sur le ruban. Dans l'état  $q_{b++}$ , la machine va ajouter deux  $b$  consécutifs, et pour cela elle passera par l'état intermédiaire  $q_{b+}$ . Enfin l'état  $q_{\|\}$  est atteint chaque fois qu'un mot de la forme  $a^n b^n$  est inscrit sur le ruban. Un symbole de séparation  $\|$  est inscrit sur le ruban à la fin de chaque mot de la forme  $a^n b^n$ . La fonction de transition  $\delta$  est

donnée par les règles suivantes :

$$\left. \begin{array}{l} \delta_{\text{ex}}(q_0, \#) = (q_{\text{b}^+}, a, \rightarrow) \\ \delta_{\text{ex}}(q_{\text{b}^+}, \#) = (q_{\parallel}, b, \rightarrow) \\ \delta_{\text{ex}}(q_{\parallel}, \#) = (q_{\parallel}, \parallel, \cdot) \end{array} \right\} \text{Initialisation du ruban : la machine écrit le premier mot } ab \text{ sur le ruban et positionne sa tête de calcul sur le symbole de séparation } \parallel \text{ à la droite du mot } ab.$$

$$\left. \begin{array}{l} \delta_{\text{ex}}(q_{\parallel}, \parallel) = (q_{\parallel}, \parallel, \leftarrow) \\ \delta_{\text{ex}}(q_{\parallel}, b) = (q_{\parallel}, b, \leftarrow) \\ \delta_{\text{ex}}(q_{\parallel}, a) = (q_{\text{a}^+}, a, \rightarrow) \end{array} \right\} \text{Si un mot } a^n b^n \parallel \text{ est écrit sur le ruban, et que la tête de calcul se trouve sur le symbole } \parallel \text{ et est dans l'état } q_{\parallel}, \text{ alors la machine recherche la lettre } a \text{ la plus à droite dans le mot } a^n b^n.$$

La machine remplace le lettre  $b$  la plus à gauche par une lettre  $a$  puis cherche le symbole de séparation  $\parallel$  à droite du mot. Une fois ce symbole atteint, la machine le remplace par le mot  $bb\parallel$ , de sorte que le mot  $a^{n+1}b^{n+1}\parallel$  est à présent écrit sur le ruban, puis la tête de calcul se place sur le symbole  $\parallel$  et passe dans l'état  $q_{\parallel}$ .

Un calcul de cette machine (qui débute par le mot vide sur le ruban) passera nécessairement par les configurations du ruban de la Figure 4.2 (où le temps s'écoule de bas en haut).

...	...	...	...	...	...	...	...	...	...	...	...	...
...	#	a	a	a	a	b	(q <sub>b<sup>+</sup></sub> , b)		#	#	...	...
...	#	a	a	a	a	(q <sub>b<sup>+</sup></sub> , b)	b		#	#	...	...
...	#	a	a	(q <sub>  </sub> , a)	b	b	b		#	#	...	...
...	#	a	a	a	(q <sub>  </sub> , b)	b	b		#	#	...	...
...	#	a	a	a	b	(q <sub>  </sub> , b)	b		#	#	...	...
...	#	a	a	a	b	b	(q <sub>  </sub> , b)		#	#	...	...
...	#	a	a	a	b	b	b	(q <sub>  </sub> ,   )	#	#	...	...
...	#	a	a	a	b	b	b	(q <sub>  </sub> , #)	#	#	...	...
...	#	a	a	a	b	b	(q <sub>b<sup>+</sup></sub> , #)	#	#	#	...	...
...	#	a	a	a	b	(q <sub>b<sup>+</sup></sub> ,   )	#	#	#	#	...	...
...	#	a	a	(q <sub>a<sup>+</sup></sub> , b)	b		#	#	#	#	...	...
...	#	a	(q <sub>  </sub> , a)	b	b		#	#	#	#	...	...
...	#	a	a	(q <sub>  </sub> , b)	b		#	#	#	#	...	...
...	#	a	a	b	(q <sub>  </sub> , b)		#	#	#	#	...	...
...	#	a	a	b	b	(q <sub>  </sub> ,   )	#	#	#	#	...	...
...	#	a	a	b	b	(q <sub>  </sub> , #)	#	#	#	#	...	...
...	#	a	a	b	(q <sub>b<sup>+</sup></sub> , #)	#	#	#	#	#	...	...
...	#	a	(q <sub>b<sup>+</sup></sub> ,   )	#	#	#	#	#	#	#	...	...
...	#	a	(q <sub>a<sup>+</sup></sub> , b)		#	#	#	#	#	#	...	...
...	#	(q <sub>  </sub> , a)	b		#	#	#	#	#	#	...	...
...	#	a	(q <sub>  </sub> , b)		#	#	#	#	#	#	...	...
...	#	a	b	(q <sub>  </sub> ,   )	#	#	#	#	#	#	...	...
...	#	a	b	(q <sub>  </sub> , #)	#	#	#	#	#	#	...	...
...	#	a	(q <sub>b<sup>+</sup></sub> , #)	#	#	#	#	#	#	#	...	...
...	#	(q <sub>0</sub> , #)	#	#	#	#	#	#	#	#	...	...

FIGURE 4.2 – Exemple de calcul d'une machine de Turing qui énumère sur son ruban tous les mots de la forme  $a^n b^n$  pour tout  $n \in \mathbb{N}$ .

Le comportement d'une machine de Turing, c'est-à-dire son ensemble de règles, est codé dans un sous-shift de type fini dont on donne les motifs autorisés. La dimension horizontale joue le rôle du ruban de la machine, tandis que la dimension verticale représente l'évolution du temps (qui s'écoule de bas en haut). On obtient ainsi les diagrammes espace-temps de la machine  $\mathcal{M}$ , que l'on peut construire à l'aide des motifs autorisés de taille  $3 \times 2$  suivants :

- si le motif code une partie du ruban dans laquelle la tête de calcul n'apparaît pas, les deux

lignes du motif sont identiques et pour  $x, y, z \in \Gamma$  on autorise le motif :

$x$	$y$	$z$
$x$	$y$	$z$

- si la tête de calcul est présente dans la partie du ruban, on code les règles de transition de la machine. Par exemple la règle  $\delta(q_1, x) = (q_2, y, \leftarrow)$  sera codée par les motifs :

$v$	$w$	$(q_2, z)$
$v$	$w$	$z$

$w$	$(q_2, z)$	$y$
$w$	$z$	$(q_1, x)$

$(q_2, z)$	$y$	$z'$
$z$	$(q_1, x)$	$z'$

$y$	$z'$	$z''$
$(q_1, x)$	$z'$	$z''$

- enfin si  $q_F$  est un état final, alors la machine arrête son calcul ce que l'on traduit par le motif :

$z$	$(q_F, x)$	$z'$
$z$	$(q_F, x)$	$z'$

On note  $P_{\mathcal{M}}$  l'ensemble des motifs interdits sur l'alphabet  $\mathcal{A}_{\mathcal{M}} = \Gamma \cup (Q \times \Gamma)$  construit à partir des règles de transition de  $\mathcal{M}$ , c'est-à-dire les motifs  $3 \times 2$  qui ne sont pas représentés dans la liste ci-dessus.

Considérons à présent le sous-shift de type fini défini à partir des motifs interdits  $\mathcal{F}_{\mathcal{M}} = \mathcal{A}_{\mathcal{M}}^{[0,2] \times [0,1]} \setminus P_{\mathcal{M}}$ . Il contient tous les diagrammes espace-temps de la machine  $\mathcal{M}$ , mais aussi d'autres configurations qui ne sont jamais atteintes par la machine. Avec la machine de Turing de l'exemple précédent, le SFT  $\mathbf{T}(\mathcal{A}_{\mathcal{M}_{\text{ex}}}, 2, \mathcal{F}_{\mathcal{M}_{\text{ex}}})$  contient une configuration dans laquelle le ruban est dans l'état suivant,

...	#	...	#	$a$	$b$	$b$	$b$	$b$	$b$	$(q_{\parallel}, \parallel)$	#	...	#	...
-----	---	-----	---	-----	-----	-----	-----	-----	-----	------------------------------	---	-----	---	-----

état du ruban qui est incohérent puisque jamais atteint dans un calcul de  $\mathcal{M}_{\text{ex}}$ .

Le problème vient du manque d'information sur le début d'un calcul. Il faut spécifier un point dans  $\mathbb{Z}^2$  qui jouera le rôle d'origine du calcul : à cette position la tête de calcul est dans l'état initial  $q_0$ , placée au début du mot d'entrée, en dehors duquel le ruban ne contient que des symboles blanc #. Ainsi on s'assure que la configuration du sous-shift  $\mathbf{T}_{\mathcal{F}_{\mathcal{M}}}$  produite avec cette contrainte correspond bien à un diagramme espace-temps de la machine  $\mathcal{M}$ .

### 4.3 Problème de complétion

On va s'intéresser au problème de décision suivant :

“Étant donné un jeu fini de tuile de Wang et une tuile  $t$ ,  
est-il possible de paver le plan sachant que la tuile  $t$  est placée à l'origine?”

Soit  $\mathcal{A}_{\bullet} = \{\bullet, \leftarrow, \rightarrow, \uparrow, \downarrow\}$ , on considère le SFT suivant :

$$\mathbf{T}(\mathcal{A}_{\bullet}, 2, \mathcal{F}_{\bullet}) = \overline{\mathcal{O}(x)} \text{ où } x = \begin{array}{cccccccc} \dots & \vdots & \dots \\ \dots & \downarrow & \dots \\ \dots & \downarrow & \dots \\ \dots & \rightarrow & \rightarrow & \rightarrow & \bullet & \leftarrow & \leftarrow & \leftarrow & \dots \\ \dots & \uparrow & \dots \\ \dots & \uparrow & \dots \\ \dots & \vdots & \dots \end{array}$$

Soit  $\mathbf{T} = \mathbf{T}(\mathcal{A}_{\bullet} \times \mathcal{A}_{\mathcal{M}}, 2, \mathcal{F}_{\bullet} \cup \mathcal{F}_{\mathcal{M}} \cup \mathcal{F}_{\text{Synchrono}} \cup \{q_F\})$  où  $\mathcal{F}_{\text{Synchrono}}$  synchronise :

- $\bullet$  avec  $(q_0, \#)$ ,
- $\uparrow, \rightarrow$  et  $\leftarrow$  avec  $\#$

$(\bullet, (q_0, \#))$  se complète en une configuration de  $\mathbf{T} \iff \mathcal{M}$  ne s'arrête pas

**Théorème 4.3.1** ([Wan61]). *Le problème de complétion est indécidable.*

Comment introduire une origine dans un sous-shift? Par compacité du sous-shift, il est impossible d'imposer qu'un symbole origine apparaisse exactement qu'une seule fois dans chaque configuration du sous-shift.

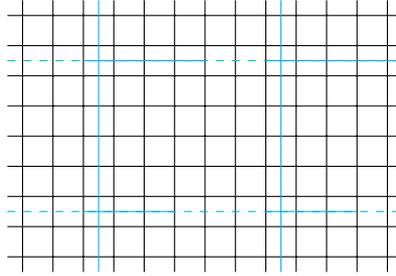
## 4.4 Problème du pavage périodique

L'indécidabilité semblant venir de la quasipériodicité, il est naturel de se demander s'il est possible de décider si un jeu fini de dominos pave périodiquement le plan ou non – c'est le problème périodique des dominos, prouvé indécidable par Gurevich et Koryakov (1972).

**Théorème 4.4.1** ([GK72]). *Le problème du pavage périodique, c'est à dire se demander si un SFT donné admet un pavage périodique, est indécidable.*

*Démonstration.* On suppose qu'il existe un SFT  $\mathbf{T}_{\text{Apé}} = \mathbf{T}(\mathcal{A}_{\text{Apé}}, 2, \mathcal{F}_{\text{Apé}}) \neq \emptyset$  qui n'a pas de configuration périodique.

On considère le sous-shift sur  $\mathcal{A}_{\text{Line}} = \left\{ \begin{array}{c} \square \\ \square \\ \square \\ \square \\ \square \end{array} \right\}$  ainsi que les règles de type fini  $\mathcal{F}_{\text{Line}}$  permettant l'alignement des lignes ainsi que l'alternance de  $\square$  et  $\square$  entre deux coins. On obtient des configurations avec des lignes partageant les configurations en des rectangles arbitrairement grands du type :



Étant donné une machine de Turing  $\mathcal{M}$  et le sous-shift de type fini associé  $\mathbf{T}(\mathcal{A}_{\mathcal{M}}, 2, \mathcal{F}_{\mathcal{M}})$ , on construit le sous-shift de type  $\mathbf{T}_{\text{Final}} = \mathbf{T}(\mathcal{A}_{\text{Apé}} \times \mathcal{A}_{\text{Line}} \times \mathcal{A}_{\mathcal{M}}, 2, \mathcal{F}_{\text{Line}} \cup \widetilde{\mathcal{F}_{\text{Apé}}} \cup \widetilde{\mathcal{F}_{\mathcal{M}}} \cup \mathcal{F}_{\text{Syncro}})$  où

- $\widetilde{\mathcal{F}_{\text{Apé}}}$  vérifie que la condition imposée par  $\mathcal{F}_{\text{Apé}}$  est vérifiée s'il n'y a pas de ligne sur la couche  $\mathcal{A}_{\text{Line}}$ ,
- $\widetilde{\mathcal{F}_{\mathcal{M}}}$  vérifie que la condition imposée par  $\mathcal{F}_{\mathcal{M}}$  est vérifiée s'il n'y a pas de ligne sur la couche  $\mathcal{A}_{\text{Line}}$ ,
- $\mathcal{F}_{\text{Syncro}}$  vérifie qu'il y a l'état  $(q_0, \#)$  sur le coin en bas à gauche,  $\#$  sur toute la ligne du bas et que  $q_F$  apparaît dans la ligne supérieure (on utilise le fait que la transition  $\begin{array}{c} \square \\ \square \end{array}$  doit apparaître sur le côté horizontal du rectangle).

Le SFT  $\mathbf{T}_{\text{Final}}$  admet alors une configuration périodique si et seulement si il existe un pas de grille où l'état final de  $\mathcal{M}$  est atteint, c'est à dire  $\mathcal{M}$  s'arrête.  $\square$

## 4.5 Le problème du domino

### 4.5.1 Un peu d'histoire

**Théorème 4.5.1.** *Le problème du domino, c'est à dire de se demander "si étant donné un ensemble de motif  $\mathcal{F}$  on a  $\mathbf{T}(\mathcal{A}, 2, \mathcal{F}) \neq \emptyset$ ", est indécidable.*

Il existe plusieurs preuves pour démontrer ce résultat. La première réponse est donnée par R. Berger [Ber66] pour répondre à une question posée par Wang [Wan61] qui était son directeur de thèse. Cette preuve un peu technique se base sur un pavage apériodique qui comporte 20 426 tuiles. Suivant la même idée, R. M. Robinson [Rob71] propose une preuve basée sur son pavage apériodique. L'idée est que chaque super-tuile définit un espace de calcul qui permet de simuler une machine de Turing donnée. Il suffit d'interdire les états finaux, ainsi on peut paver le plan si et seulement si la machine Turing s'arrête.

J. Kari propose une preuve originale basée sur sa construction de pavage à 13 tuiles [Kar07]. Il réduit le problème du domino à un problème d'immortalité sur les fonctions affines par morceau du à Hooper [Hoo66] (un autre étudiant de Wang). L'avantage de cette preuve est qu'elle s'adapte à des pavages sur d'autres espaces tel que le plan hyperbolique.

Dans ce cours, nous allons voir une preuve d'indécidabilité qui se rapproche de celle proposée par Robinson. Les calculs vont se dérouler sur des bandes infinies positionnées à l'aide d'un pavage substitutif, le calcul étant initialisé par un système d'horloge.

#### 4.5.2 Espaces de calcul pour MT

Nous allons présenter une construction d'espaces de calcul pour des machines de Turing. Rappelons que les machines considérées travaillent sur un ruban semi-infini (la tête de lecture ne peut pas se déplacer à gauche de sa position initiale). Dans cette partie, un espace de calcul pour une machine de Turing  $\mathcal{M}$  est un rectangle dans le plan discret. On cherche à construire un sous-shift  $\mathbf{T}_{\text{calcul}}$  avec les propriétés suivantes :

- $\mathbf{T}_{\text{calcul}}$  est sofique ;
- toute configuration de  $\mathbf{T}_{\text{calcul}}$  contient des espaces de calcul de taille arbitrairement grande, en espace et en temps.

Un tel sous-shift assure de voir apparaître les diagrammes espace-temps de la machine  $\mathcal{M}$ .

Pour simplifier les notations, on introduit deux opérations sur les sous-shifts, ces opérations seront étudiées plus précisément dans le chapitre suivant :

- étant donnés deux sous-shift  $\mathbf{T}$  et  $\mathbf{T}'$ , on définit le produit  $\mathbf{Prod}(\mathbf{T}, \mathbf{T}') = \mathbf{T} \times \mathbf{T}'$  ;
- et étant donnés un sous-shift  $\mathbf{T}$  et un ensemble fini de motif  $\mathcal{F}$ , on définit le sous-shift construit à partir de  $\mathbf{T}$  auquel on interdit les motifs  $\mathcal{F}$  :

$$\mathbf{TF}_{\mathcal{F}}(\mathbf{T}) = \mathbf{T} \setminus \left( \bigcup_{\mathbf{i} \in \mathbb{Z}^d, u \in \mathcal{F}} \sigma^{\mathbf{i}}([u]) \right).$$

On choisit ici d'utiliser un sous-shift substitutif pour définir les espaces de calcul. Soit l'alphabet  $\mathcal{G}_1$  à quatre éléments présenté dans la Figure 4.3. Cet alphabet  $\mathcal{G}_1 = \{ \square, \blacksquare, \blacktriangleright, \blacktriangleleft \}$  peut être divisé en deux parties. Les lettres  $\blacksquare$ ,  $\blacktriangleright$  et  $\blacktriangleleft$  correspondent aux *tuiles de calcul*, où les calculs de machine de Turing seront effectivement implémentés, tandis que la lettre  $\square$  correspond aux *tuiles de communication* par lesquelles de l'information peut circuler. Plus précisément  $\blacktriangleright$  et  $\blacktriangleleft$  sont des *tuiles de calcul de bord*. Sur cet alphabet  $\mathcal{G}_1$  on définit une substitution  $s_1$  par les règles données dans la Figure 4.3.



FIGURE 4.3 – Règles de la substitution  $s_1$ .

**Proposition 4.5.2.** *La substitution  $s_1$  est à dérivation unique.*

*Démonstration.* Le motif  $\begin{array}{|c|} \hline \ast \\ \hline \end{array}$  apparaît dans chacune des règles de la substitution  $s_1$ . Pour toute configuration  $x \in \mathbf{T}_{\{s_1\}}$ , il existe donc  $(i, j) \in [0, 3] \times [0, 1]$  tel que  $x_{\{n_1+i+1\} \times [n_2+j+1, n_2+j+2]} = \begin{array}{|c|} \hline \ast \\ \hline \end{array}$  pour tout  $(n_1, n_2) \in \mathbb{N} \times \mathbb{N}$ . De plus, ce motif ne peut pas apparaître à une autre position, donc le couple d'entier  $(i, j)$  est choisi de manière unique. Considérons la partition du plan  $([n_1 + i, n_1 + i + 3] \times [n_2 + j, n_2 + j + 1])_{(n_1, n_2) \in 4\mathbb{N} \times 2\mathbb{N}}$  pour la configuration  $x$ . Comme toutes les tuiles ont des images différentes par la substitution  $s_1$ , cette partition donne un unique antécédent  $y$  à  $x$  par  $s_1$ . On en déduit que  $s_1(y) = \sigma^{(i,j)}(x)$ ; la substitution  $s_1$  est donc bien à dérivation unique.  $\square$

Cette substitution  $s_1$  définit un sous-shift  $\mathbf{T}_{\{s_1\}}$ , sofique en vertu du Théorème 3.2.2. On décrit à présent comment des zones de calcul apparaissent dans les configurations de ce sous-shift. Dans cette construction, l'espace est représenté par la coordonnée horizontale, tandis que le temps est représenté par la coordonnée verticale, et s'écoule du bas vers le haut. Observons tout d'abord les motifs obtenus en itérant  $s_1$  sur une lettre (voir la Figure 4.4).

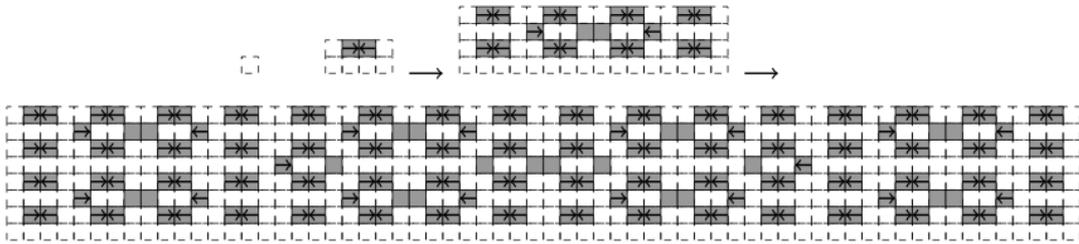


FIGURE 4.4 – Trois itérations de la substitution  $s_1$  en partant de la lettre  $\begin{array}{|c|} \hline \ast \\ \hline \end{array}$ .

Sur une ligne horizontale, une *zone de calcul* est constituée d'un groupe de tuiles de calcul  $\begin{array}{|c|} \hline \blacksquare \\ \hline \end{array}$  situées entre une tuile  $\begin{array}{|c|} \hline \rightarrow \\ \hline \end{array}$  à gauche et une tuile  $\begin{array}{|c|} \hline \leftarrow \\ \hline \end{array}$  à droite. La *taille* de la zone de calcul est le nombre de tuiles de calcul ( $\rightarrow$ ,  $\blacksquare$  ou  $\leftarrow$ ) qui constituent la zone.

Considérons une configuration  $x \in \mathbf{T}_{\{s_1\}}$  ainsi qu'une zone de calcul dans  $x$ . Comme la substitution  $s_1$  est à dérivation unique, pour tout entier  $n$  il existe une manière unique de partitionner la configuration  $x$  en rectangles de taille  $4^n \times 2^n$ , de sorte que chacun de ces rectangles est de la forme  $s_1^n(a)$  pour une certaine lettre  $a \in \mathcal{G}_1$ . Il existe donc un plus petit entier  $n$  tel que la zone de calcul de  $x$  apparaît dans  $s_1^n(a)$  pour une lettre  $a \in \mathcal{G}_1$ . Cet entier minimal sera appelé le *niveau* de la zone de calcul.

En itérant la substitution  $s_1$  sur un motif de  $\mathcal{G}_1$ , on obtient des zones de calculs aussi larges qu'on le souhaite (voir la Figure 4.5 pour un exemple).

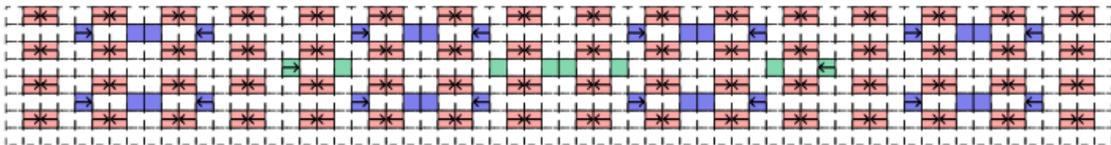


FIGURE 4.5 – Zones de calcul de différents niveaux (les zones de niveau 1 en rouge, celles de niveau 2 en bleu et celle de niveau 3 en vert) après trois itérations de  $s_1$ .

Après  $n$  itérations de la substitution  $s_1$  sur la lettre  $\begin{array}{|c|} \hline \ast \\ \hline \end{array}$  (ou n'importe quelle autre lettre de l'alphabet  $\mathcal{G}_1$ , la description étant la même à part la ligne inférieure qui change), ce qui revient à considérer le motif  $s_1^n(\begin{array}{|c|} \hline \ast \\ \hline \end{array})$ , on obtient un rectangle de taille  $4^n \times 2^n$ . Par récurrence, on peut

montrer que pour tout  $m \in [1, n]$ , on obtient  $4^{n-m} * 2^{n-m} = 8^{n-m}$  zones de calcul de niveau  $m$  dans le motif  $s^n(\begin{smallmatrix} \square \\ \square \end{smallmatrix})$ , la taille de ces zones de calcul étant  $2^m$ . Plus précisément, si sur la ligne  $j \in [0, 2^n - 1]$  de  $s^n(\begin{smallmatrix} \square \\ \square \end{smallmatrix})$  se trouve une zone de calcul de niveau  $m$ , alors sur cette ligne il y a  $4^{n-m}$  zones de calcul de niveau  $m$ . De plus pour chaque tuile de calcul située aux coordonnées  $(i, j)$ , la prochaine tuile de calcul de même niveau située dans la même colonne est séparée de celle-ci par  $2^m - 1$  tuiles de communication et se trouve donc aux coordonnées  $(i, j + 2^m)$ ; cette tuile de calcul de niveau  $m$  est dans la même position relative à sa zone de calcul que la tuile en position  $(i, j)$ . Le même phénomène se produit si l'on regarde vers le bas. L'ensemble des zones de calcul de la même taille  $2^m$  alignées verticalement forment une *bande de calcul* de taille  $2^m$  (la taille est en fait la largeur de la bande de calcul).

*Remarque.* Il est possible que sur une ligne, un symbole  $\Rightarrow$  apparait sans symbole  $\Leftarrow$  à sa droite. Cette zone de calcul dégénérée sera appelée zone de calcul infinie, et on considère qu'elle est de niveau  $+\infty$ .

**Proposition 4.5.3.** *Considérons une configuration  $x \in \mathbf{T}_{\{s_1\}}$  et  $C_n$  une zone de calcul de niveau  $n$  dans  $x$ . Supposons aussi que  $C_n$  apparaît dans la  $i^{\text{ième}}$  ligne de  $x$ , que l'on notera  $x_{\mathbb{Z} \times \{i\}}$ . Alors  $C_n$  vérifie les propriétés suivantes :*

1. *la zone de calcul  $C_n$  contient  $2^n$  tuiles de calcul, séparées par des tuiles de communication ;*
2. *sur la ligne  $x_{\mathbb{Z} \times \{i\}}$  il n'y a que des zones de calcul de niveau  $n$ , séparées les unes des autres par  $2^{2n-1}$  tuiles de communication ;*
3. *la ligne  $x_{\mathbb{Z} \times \{i\}}$  est répétée verticalement toutes les  $2^n$  lignes, autrement dît  $x_{\mathbb{Z} \times \{i\}} = x_{\mathbb{Z} \times \{i+k \times 2^n\}}$  pour tout entier  $k \in \mathbb{Z}$  ;*
4. *selon la coordonnée verticale, entre deux tuiles de calcul consécutives des lignes  $x_{\mathbb{Z} \times \{i\}}$  et  $x_{\mathbb{Z} \times \{i+k \times 2^n\}}$ , on trouve uniquement des tuiles de communication, au nombre de  $2^n - 1$ .*

Voyons à présent comment deux tuiles de calcul d'une même bande de calcul peuvent communiquer.

### 4.5.3 Communication dans le sous-shift $\mathbf{T}_{\{s_1\}}$

On décrit dans cette partie comment les tuiles de communication sont utilisées.

**Principe de communication dans un sous-shift** Un *canal de communication* est une suite de tuiles de communication adjacentes. Ce canal commence et se termine par une tuile de calcul.

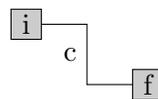


FIGURE 4.6 – Un canal de communication entre les tuiles de calcul  $i$  et  $f$ .

Un *transfert d'information* est formé de trois données :

- un canal de communication  $c$  ;
- une *tuile de calcul initiale* appelée  $i$  et une *tuile de calcul finale* appelée  $f$  ;
- des règles locales qui déterminent le symbole qui sera transféré à travers le canal de communication, qui dépend à la fois de la direction du canal partant de  $i$  (respectivement arrivant sur  $f$ ) et du symbole contenu par la tuile  $i$  (respectivement  $f$ ).

Deux tuiles de communication voisines portent le même symbole, qui est transféré le long du canal de communication. Précisons que les tuiles  $i$  et  $f$  ne contiennent pas forcément la même lettre (par exemple la règle locale peut modifier le symbole transféré à son arrivée sur la tuile  $f$ ). Une même tuile de calcul peut se trouver aux extrémités de canaux de communication différents, mais on impose ici que ce nombre de canaux soit borné.

Étant donné un sous-shift  $\mathbf{T}$  contenant des canaux de communication, il est possible de coder des transferts d'information dans un sous-shift  $\mathbf{T}_{\text{info}}$ . Si les symboles transférés dépendent uniquement d'un motif fini autour des tuiles initiales et finales de chaque canal, et que  $\mathbf{T}$  est un sous-shift de type fini (respectivement sofique), alors  $\mathbf{T}_{\text{info}}$  est aussi de type fini (respectivement sofique).

**Communication à l'intérieur d'une bande de calcul** Deux tuiles de calcul dans une même bande de niveau  $n$  situées sur une même colonne peuvent communiquer grâce aux tuiles de communication des  $2^n - 1$  lignes intermédiaires (transfert vertical de l'information). Au sein d'une même zone de calcul de niveau  $n$ , deux tuiles de calcul adjacentes (c'est-à-dire qu'entre elles deux il n'y a que des tuiles de communication) peuvent communiquer via les  $2 * 4^{n-1} - 2^n$  tuiles de communication qui les séparent (transfert horizontal de l'information). La Figure 4.7 illustre ces propos.

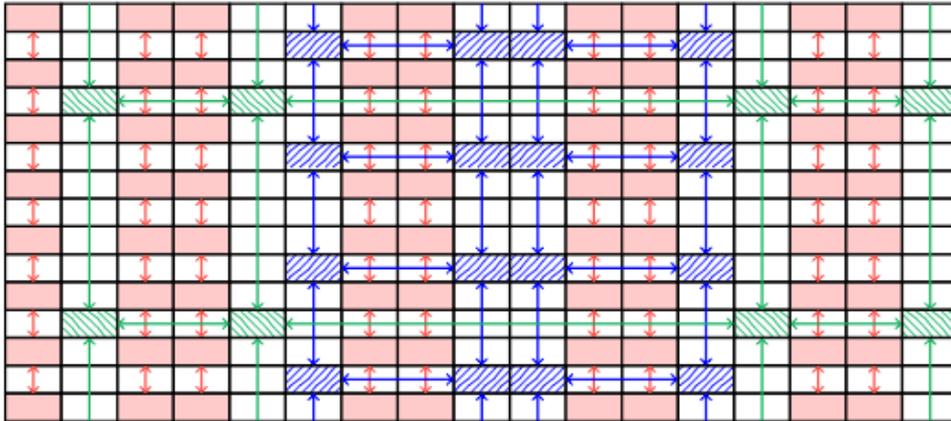


FIGURE 4.7 – Les zones de calcul de niveaux 1,2 et 3 sont signalées par des couleurs différentes. Les communications entre tuiles de calcul de ces différents niveaux sont représentées par des flèches.

Ainsi chaque tuile de communication contient une partie de deux canaux, l'un horizontal pour permettre la communication au sein d'une même zone de calcul, l'autre vertical pour permettre la communication au sein d'une même bande de calcul.

#### 4.5.4 Initialisation des calculs

Les bandes de calcul décrites dans la partie précédente sont restreintes en espace mais pas en temps. De ce fait des configurations inconsistantes de la machine de Turing peuvent apparaître. Pour résoudre ce problème, on munit chacune des bandes de calcul d'une horloge qui sera réinitialisée de façon périodique. A chaque étape de calcul, l'horloge est incrémentée et lorsqu'elle est réinitialisée, la machine de Turing recommence un nouveau calcul.

On utilise l'alphabet à quatre éléments  $\mathcal{C} = \{0, 1, \emptyset, \sim\}$  pour construire un sous-shift sofique  $\mathbf{T}_{\text{Horloge}}$  obtenu en ajoutant des règles de type fini au sous-shift sofique  $\mathbf{Prod}(\mathbf{T}_{s_1}, \mathcal{C}^{\mathbb{Z}^2})$ , où  $\mathbf{T}_{\{s_1\}}$  est le sous-shift sofique décrit dans la section 4.5.2.

On note par  $\pi_C$  la projection sur la seconde coordonnée. L'horloge est en fait un automate fini qui simule l'addition binaire modulo  $2^{2^n}$  sur un ruban de  $2^n$  tuiles. Le symbole spécial  $\emptyset$  correspond à la retenue dans l'addition binaire, et le symbole  $\sim$  est utilisé pour synchroniser des zones de calculs adjacentes de même niveau. Pour empêcher l'apparition d'états inconsistants dans l'horloge, on interdit les motifs  $\boxed{\emptyset \ 0}$ ,  $\boxed{\emptyset \ 1}$ ,  $\boxed{0 \ \emptyset}$ ,  $\boxed{x \ \sim}$  et  $\boxed{\sim \ x}$  où  $x \in \{0, 1, \emptyset\}$ ; on notera par **Consist** cette condition de type fini.

Les autres conditions de type fini **Compt** sur l'alphabet  $\mathcal{G}_1 \times \mathcal{C}$ , qui servent à décrire le processus d'addition, sont décrites dans la Figure 4.8.

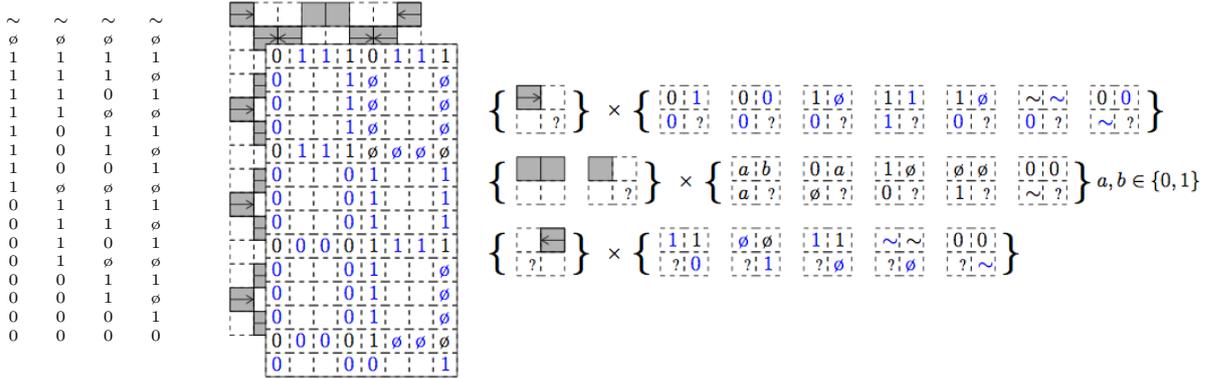


FIGURE 4.8 – À gauche, un exemple d'évolution de l'horloge sur une zone de calcul de niveau 2. Au centre l'évolution de cette horloge sur une bande de calcul de niveau 2 : les temps 001 $\emptyset$ , 0011, 01 $\emptyset\emptyset$  et 0101 sont écrits successivement. À droite certaines des règles locales pour coder l'horloge, données par les motifs autorisés.

Les horloges des différents niveaux de calcul évoluent selon les règles décrites dans la Figure 4.8, et lorsqu'un symbole  $\emptyset$  atteint la tuile de calcul la plus à gauche  $\boxed{\emptyset}$ , les horloges sont réinitialisées. Avant la réinitialisation proprement dite (retour dans la configuration dans laquelle le ruban ne contient que des symboles 0), l'horloge passe par une configuration dans laquelle son ruban ne contient que des symboles  $\sim$ . Par exemple une horloge associée à une bande de calcul de niveau 1 passera par les configurations successives 00, 01, 1 $\emptyset$ , 11,  $\emptyset\emptyset$ ,  $\sim\sim$ , 00, ... Ainsi une horloge associée à une zone de calcul de niveau  $n$  est réinitialisée après  $2^{2^n} + 2$  étapes de calcul.

Grâce à la configuration ne contenant que des symboles  $\sim$ , il est possible de synchroniser une horloge sur une bande de calcul de niveau  $n$  avec les horloges de ses deux bandes de calcul voisines de niveau  $n$ , et ce uniquement par des règles locales que l'on notera par **Synchro**. Plus précisément, ces règles locales imposent qu'une horloge soit dans la configuration  $\sim \dots \sim$  uniquement lorsque ses deux voisines sont dans la même configuration (un signal portant le symbole  $\sim$  étant envoyé par le canal de communication reliant deux bandes de calcul voisines).

Les règles de type fini **Synchro** assurent que sur une même ligne les horloges qui apparaissent, et qui correspondent donc à des bandes de calcul de même niveau, sont synchronisées (sur une même ligne, toutes les horloges sont dans la même configuration). On obtient ainsi un sous-shift sofique  $\mathbf{T}_{\text{Horloge}}$  dans lequel chaque bande de calcul du sous-shift  $\mathbf{T}_{\{s_1\}}$  est à présent munie d'une horloge. Notons qu'en aucune manière on n'impose à des horloges de bandes de calcul de niveaux différents d'être synchronisées d'aucune façon que ce soit.

$$\mathbf{T}_{\text{Horloge}} = \mathbf{TF}_{\text{Compt} \cup \text{Consist} \cup \text{Synchro}} \left( \mathbf{Prod} \left( \mathbf{T}_{\{s_1\}}, \mathcal{C}^{\mathbb{Z}^2} \right) \right)$$

**Proposition 4.5.4.** *Le sous-shift sofique  $\mathbf{T}_{\text{Horloge}}$  est tel que, à l'intérieur d'une bande de calcul de niveau  $n$  de largeur  $2^n$ , une horloge est réinitialisée toutes les  $2^{2^n} + 2$  étapes de calcul.*

#### 4.5.5 Diagramme espace-temps d'une MT dans un sous-shift sofique

Le sous-shift sofique  $\mathbf{T}_{\text{Horloge}}$  présenté dans la section 4.5.4 est ici utilisé pour construire un sous-shift sofique dans lequel apparaissent les diagrammes espace-temps d'une machine de Turing  $\mathcal{M}$ . Pour cela les règles  $P_{\mathcal{M}}$ , définies dans la section 4.2, sont ajoutées au sous-shift  $\mathbf{T}_{\text{Horloge}}$ . La difficulté est que les espaces de calcul présents dans les configurations de ce sous-shift sont fractionnés, et que l'on ne peut donc pas appliquer directement les règles locales  $P_{\mathcal{M}}$ . Il faut alors les modifier au préalable pour tenir compte des canaux de communication décrits dans la section 4.5.2.

Partant du sous-shift sofique  $\mathbf{T}_{\text{Horloge}}$  défini dans la Partie 4.5.4, on l'insère dans le sous-shift  $\mathbf{Prod}(\mathbf{T}_{\text{Horloge}}, \tilde{\mathcal{A}}^{\mathbb{Z}^2})$  où le nouvel alphabet  $\tilde{\mathcal{A}}$  est défini de la manière suivante :  $\tilde{\mathcal{A}} = \mathcal{A}_{\mathcal{M}} \cup (\mathcal{A}_{\mathcal{M}} \times \mathcal{A}_{\mathcal{M}} \times \mathcal{A}_{\mathcal{M}})$ . Un symbole de  $\tilde{\mathcal{A}}$  représente donc soit un symbole de  $\mathcal{A}_{\mathcal{M}}$  sur une tuile de calcul, soit la superposition de trois symboles de  $\mathcal{A}_{\mathcal{M}}$  transférés (horizontalement pour le premier et le deuxième, verticalement pour le troisième) par une tuile de communication. On définit  $\pi_{\mathcal{G}_1}$  et  $\pi_{\mathcal{C}}$  les projections respectivement sur  $\mathcal{G}_1$  et  $\mathcal{C}$  pour la première coordonnée de  $\mathbf{Prod}(\mathbf{T}_{\text{Horloge}}, \tilde{\mathcal{A}}^{\mathbb{Z}^2})$ . De plus on appellera  $\pi_{\tilde{\mathcal{A}}}$  la projection sur la deuxième coordonnée de  $\mathbf{Prod}(\mathbf{T}_{\text{Horloge}}, \tilde{\mathcal{A}}^{\mathbb{Z}^2})$ . Pour une tuile de communication, on peut ainsi écrire  $\pi_{\tilde{\mathcal{A}}_1}$ ,  $\pi_{\tilde{\mathcal{A}}_2}$  et  $\pi_{\tilde{\mathcal{A}}_3}$  pour respectivement la première, deuxième et troisième coordonnées de  $\mathcal{A}_{\mathcal{M}} \times \mathcal{A}_{\mathcal{M}} \times \mathcal{A}_{\mathcal{M}}$ .

On ajoute des conditions de type fini au sous-shift sofique  $\mathbf{Prod}(\mathbf{T}_{\text{Horloge}}, \tilde{\mathcal{A}}^{\mathbb{Z}^2})$ . Ces conditions s'expriment par des motifs dont le support est de la forme

$$\begin{array}{|c|c|c|} \hline & a & \\ \hline b & c & d \\ \hline & e & \\ \hline \end{array} \quad \text{avec } a, b, c, d, e \in \mathcal{G}_1 \times \mathcal{C} \times \tilde{\mathcal{A}}$$

Les conditions sont les suivantes :

- si la tuile centrale correspond à une tuile de communication dans  $\mathbf{T}_{\text{Horloge}}$ , autrement dit  $\pi_{\mathcal{G}_1}(c) = \lfloor \_ \rfloor$ , on applique l'ensemble de conditions **Transfert** : les première et deuxième coordonnées sont constantes le long de la ligne centrale, tandis que la troisième coordonnée est constante le long de la colonne centrale. Plus précisément,  $\pi_{\tilde{\mathcal{A}}_1}(b) = \pi_{\tilde{\mathcal{A}}_1}(c) = \pi_{\tilde{\mathcal{A}}_1}(d)$ ,  $\pi_{\tilde{\mathcal{A}}_2}(b) = \pi_{\tilde{\mathcal{A}}_2}(c) = \pi_{\tilde{\mathcal{A}}_2}(d)$  et  $\pi_{\tilde{\mathcal{A}}_3}(a) = \pi_{\tilde{\mathcal{A}}_3}(c) = \pi_{\tilde{\mathcal{A}}_3}(e)$ . Ces conditions sont valables si toutes les tuiles adjacentes sont des tuiles de communication. Si une tuile de calcul est présente dans le voisinage, on utilise la projection  $\pi_{\tilde{\mathcal{A}}}$  ;
- si la tuile centrale correspond à une tuile de calcul dans  $\mathbf{T}_{\text{Horloge}}$ , autrement dit  $\pi_{\mathcal{G}_1}(c) \in \{\blacksquare, \blacktriangleright, \blacktriangleleft\}$ , on utilise l'une des conditions suivantes selon le cas :
  - conditions **Init** : lorsque l'horloge est dans son état initial  $0 \dots 0$ , chaque tuile contient un symbole blanc  $\sharp$  et la tête de calcul est dans l'état initial  $q_0$ , positionnée sur la tuile de calcul  $\blacktriangleright$ , ce que l'on exprime par :
    - si  $\pi_{\mathcal{C}}(c) = \sim$  et  $\pi_{\mathcal{G}_1}(c) = \blacktriangleright$  alors  $\pi_{\tilde{\mathcal{A}}}(c) = \pi_{\tilde{\mathcal{A}}_1}(d) = \pi_{\tilde{\mathcal{A}}_2}(b) = \pi_{\tilde{\mathcal{A}}_3}(a) = (q_0, \sharp)$ ,
    - si  $\pi_{\mathcal{C}}(c) = \sim$  et  $\pi_{\mathcal{G}_1}(c) \in \{\blacksquare, \blacktriangleleft\}$  alors  $\pi_{\tilde{\mathcal{A}}}(c) = \pi_{\tilde{\mathcal{A}}_1}(d) = \pi_{\tilde{\mathcal{A}}_2}(b) = \pi_{\tilde{\mathcal{A}}_3}(a) = \sharp$  ;
  - conditions **Comp** : si l'horloge n'est pas dans son état initial, on utilise les règles décrites dans  $\mathcal{P}_{\mathcal{M}}$ . Plus précisément :
    - si  $\pi_{\mathcal{C}}(c) \neq \sim$  et  $\pi_{\mathcal{G}_1}(c) = \blacksquare$  alors

$$\begin{array}{|c|c|c|} \hline & \pi_{\tilde{\mathcal{A}}_3}(a) & \\ \hline \pi_{\tilde{\mathcal{A}}_1}(b) & \pi_{\tilde{\mathcal{A}}}(c) & \pi_{\tilde{\mathcal{A}}_2}(d) \\ \hline \end{array} \in P_{\mathcal{M}}, \quad \pi_{\tilde{\mathcal{A}}}(c) = \pi_{\tilde{\mathcal{A}}_2}(b) = \pi_{\tilde{\mathcal{A}}_1}(d) \text{ et } \pi_{\tilde{\mathcal{A}}}(c) = \pi_{\tilde{\mathcal{A}}_3}(e),$$

- si  $\pi_C(c) \neq \sim$ ,  $\pi_{G_1}(c) = \blacktriangleright$  et si la troisième coordonnées de  $\delta(\pi_{\tilde{A}}(c))$  n'est pas  $\leftarrow$ , c'est-à-dire que la fonction de transition de la machine de Turing ne déplace pas la tête de calcul vers la gauche, alors

$$\begin{array}{|c|c|c|} \hline & \pi_{\tilde{A}_3}(a) & \\ \hline \# & \pi_{\tilde{A}}(c) & \pi_{\tilde{A}_2}(d) \\ \hline \end{array} \in P_{\mathcal{M}}, \quad \pi_{\tilde{A}}(c) = \pi_{\tilde{A}_2}(b) = \pi_{\tilde{A}_1}(d) \text{ and } \pi_{\tilde{A}}(c) = \pi_{\tilde{A}_3}(e),$$

- si  $\pi_C(c) \neq \sim$ ,  $\pi_{G_1}(c) = \blacktriangleleft$  et si la troisième coordonnée de  $\delta(\pi_{\tilde{A}}(c))$  n'est pas  $\rightarrow$ , c'est-à-dire que la fonction de transition de la machine de Turing ne déplace pas la tête de calcul vers la droite, alors

$$\begin{array}{|c|c|c|} \hline & \pi_{\tilde{A}_3}(a) & \\ \hline \pi_{\tilde{A}_1}(b) & \pi_{\tilde{A}}(c) & \# \\ \hline \end{array} \in P_{\mathcal{M}}, \quad \pi_{\tilde{A}}(c) = \pi_{\tilde{A}_2}(b) = \pi_{\tilde{A}_1}(d) \text{ et } \pi_{\tilde{A}}(c) = \pi_{\tilde{A}_3}(e);$$

- conditions **Bord** : si la tête de calcul veut se déplacer à gauche d'une tuile de calcul  $\blacktriangleright$  ou à droite d'une tuile de calcul  $\blacktriangleleft$ , elle entre dans un état spécial  $q_{\text{wait}}$  et le calcul est suspendu jusqu'à la prochaine réinitialisation de l'horloge. Plus précisément :
  - si  $\pi_C(c) \neq \sim$ ,  $\pi_{G_1}(c) = \blacktriangleright$  et si la troisième coordonnée de  $\delta(\pi_{\tilde{A}}(c))$  est  $\leftarrow$ , alors

$$\begin{array}{|c|c|c|} \hline q_{\text{wait}} & & \\ \hline \pi_{\tilde{A}}(c) & \pi_{\tilde{A}_2}(d) & \\ \hline \end{array}, \quad \pi_{\tilde{A}}(c) = \pi_{\tilde{A}_2}(b) = \pi_{\tilde{A}_1}(d) \text{ and } \pi_{\tilde{A}}(c) = \pi_{\tilde{A}_3}(e);$$

- si  $\pi_C(c) \neq \sim$ ,  $\pi_{G_1}(c) = \blacktriangleleft$  et si la troisième coordonnée de  $\delta(\pi_{\tilde{A}}(c))$  est  $\rightarrow$ , alors

$$\begin{array}{|c|c|c|} \hline & q_{\text{wait}} & \\ \hline \pi_{\tilde{A}_1}(b) & \pi_{\tilde{A}}(c) & \\ \hline \end{array}, \quad \pi_{\tilde{A}}(c) = \pi_{\tilde{A}_2}(b) = \pi_{\tilde{A}_1}(d) \text{ and } \pi_{\tilde{A}}(c) = \pi_{\tilde{A}_3}(e);$$

- si  $\pi_C(c) \neq \sim$ ,  $\pi_{G_1}(c) \in \{ \blacksquare, \blacktriangleright, \blacktriangleleft \}$  et  $\pi_{\tilde{A}} = q_{\text{wait}}$ , alors

$$\pi_{\tilde{A}}(c) = \pi_{\tilde{A}_3}(a) = \pi_{\tilde{A}_3}(e) = \pi_{\tilde{A}_2}(b) = \pi_{\tilde{A}_1}(d) = q_{\text{wait}},$$

- si  $\pi_C(c) \neq \sim$  et  $\pi_{\tilde{A}_1}(b) = q_{\text{wait}}$  ou bien  $\pi_{\tilde{A}_2}(d) = q_{\text{wait}}$  alors  $\pi_{\tilde{A}}(c) = q_{\text{wait}}$ .

Ces conditions de type fini permettent de définir le sous-shift sofique  $\widetilde{\mathbf{T}}_{\mathcal{M}}$  de la manière suivante :

$$\widetilde{\mathbf{T}}_{\mathcal{M}} = \mathbf{TF}_{\text{Transfert} \cup \text{Init} \cup \text{Comp} \cup \text{Bord}} \left( \mathbf{Prod} \left( \mathbf{T}_{\text{Horloge}}, \tilde{\mathcal{A}}^{\mathbb{Z}^2} \right) \right).$$

Pour plus de clarté, toutes les conditions de type fini **Transfert**, **Init**, **Comp** et **Bord** sont regroupées en une seule condition **Work** $_{\mathcal{M}}$ . La construction se résume ainsi en :  $\widetilde{\mathbf{T}}_{\mathcal{M}} = \mathbf{TF}_{\text{Work}_{\mathcal{M}}} \left( \mathbf{Prod} \left( \mathbf{T}_{\text{Grid}}, \mathcal{A}_{\text{Comp}(\mathcal{M})}^{\mathbb{Z}^2} \right) \right)$  pour toute machine de Turing  $\mathcal{M}$ .

Sur chaque bande de calcul apparaissent ainsi des morceaux du diagramme espace-temps de la machine de Turing  $\mathcal{M}$ , avec pour mot d'entrée le mot vide. Chacun de ces morceaux de diagramme espace-temps est limité à la fois en espace, par la taille de la bande de calcul, et en temps, borné exponentiellement en la taille de la bande de calcul. Il est donc possible de trouver dans le sous-shift sofique  $\widetilde{\mathbf{T}}_{\mathcal{M}}$  des fragments arbitrairement grands du diagramme espace-temps de la machine  $\mathcal{M}$ , ce qui montre le résultat principal de cette partie :

**Théorème 4.5.5.** *Pour toute machine de Turing  $\mathcal{M}$  avec ruban semi-infini, il existe un sous-shift sofique  $\widetilde{\mathbf{T}}_{\mathcal{M}}$  qui contient les diagrammes espace-temps de taille  $2^n \times (2^{2^n} + 2)$  du calcul de la machine  $\mathcal{M}$  sur le mot vide, pour tout entier  $n$  ( $2^n$  tuiles de calcul et  $2^{2^n} + 2$  étapes de calcul).*

*Exemple 4.5.1.* Dans cet exemple la machine  $\mathcal{M}_{\text{ex}}$  commence son énumération par le mot  $ab$ . La Figure 4.9 décrit comment le calcul de la machine est codé dans la grille de calcul. Par exemple les tuiles de calcul de niveau 2 forment une zone de calcul de taille  $4 \times 18$  (ruban à quatre cellules et dix-huit étapes de calcul).

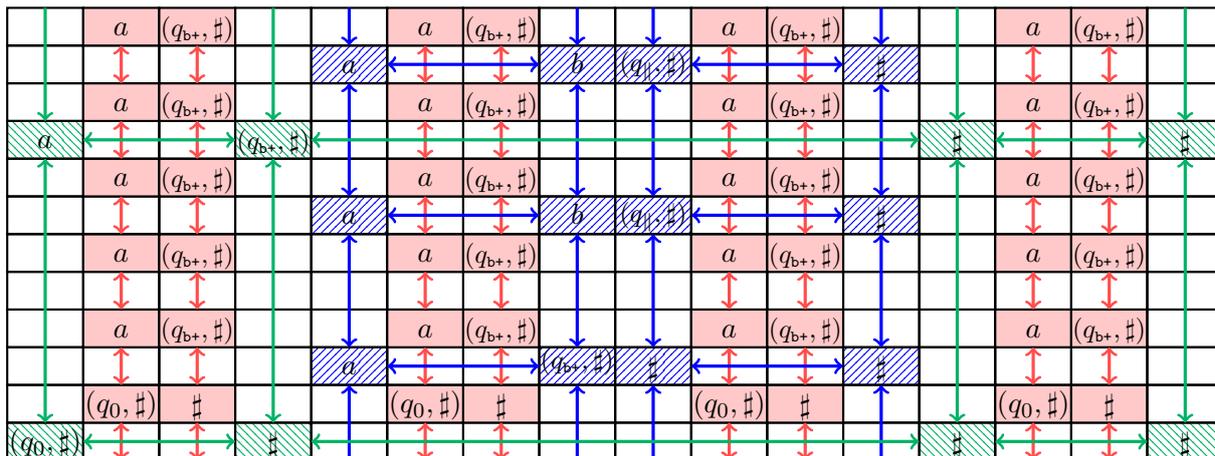


FIGURE 4.9 – Calculs d'une machine de Turing sur une grille de calcul dans laquelle apparaissent des zones de calcul de niveaux 1, 2 et 3. Il est à noter que chaque symbole  $\uparrow$  ou  $\leftrightarrow$  devrait en fait transférer deux symboles, mais ceux-ci ont ici été omis pour plus de lisibilité. Pour la même raison les états de l'horloge n'apparaissent pas.

#### 4.5.6 Retour au problème du domino

A toute machine de Turing on associe le sous-shift  $\widetilde{\mathbf{T}}_{\mathcal{M}}$  auquel on a supprimé les états finaux de la machine de Turing  $\mathcal{M}$ . Ce sous-shift est vide si et seulement si la machine de Turing  $\mathcal{M}$  s'arrête. On en déduit que le problème du domino est indécidable.

# Chapitre 5

## Opérations dynamiques sur un sous-shift

### 5.1 Opérations et simulations

On s'intéresse à différentes opérations issues de l'étude des systèmes dynamiques pour transformer un sous-shift sur  $\mathbb{Z}^d$  en un autre sous-shift.

#### 5.1.1 Simulation et classes stables

Une *classe d'opérations* sur l'ensemble des sous-shifts est un ensemble de fonctions  $op_\ell : \mathbb{S} \rightarrow \mathbb{S}$  ou  $op_\ell : \mathbb{S} \times \mathbb{S} \rightarrow \mathbb{S}$  dépendant d'un paramètre  $\ell$ . Dans ces définitions rien n'impose qu'un sous-shift et son image soient définis sur le même alphabet, ni qu'ils aient la même dimension. Ces opérations sont souvent issues de l'étude des systèmes dynamiques.

Étant donné un ensemble  $Op$  de classes d'opérations agissant sur les sous-shifts, on dit qu'un sous-shift  $\mathbf{T}$  *simule* un sous-shift  $\mathbf{T}'$  si on peut obtenir  $\mathbf{T}'$  en partant de  $\mathbf{T}$  et en appliquant un nombre fini d'opérations choisies parmi  $Op$ . On note  $\mathbf{T}' \leq_{Op} \mathbf{T}$ . Ainsi  $Cl_{Op}(\mathbf{T}) = \{\mathbf{T}' : \mathbf{T}' \leq_{Op} \mathbf{T}\}$  est le plus petit ensemble stable par  $Op$  qui contient  $\mathbf{T}$ . La *clôture* d'une classe de sous-shift  $\mathcal{U}$  pour l'ensemble d'opérations  $Op$ , noté  $Cl_{Op}(\mathcal{U})$ , est le plus petit ensemble stable par  $Op$  qui contient  $\mathcal{U}$ . Une classe de sous-shifts  $\mathcal{U}$  est *stable* pour l'ensemble d'opérations  $Op$  si  $Cl_{Op}(\mathcal{U}) = \mathcal{U}$ .

#### 5.1.2 Exemples d'opérations et résultats classiques de simulation

**Opération type fini (TF) :** Cette classe d'opérations consiste à ajouter un nombre fini de motifs interdits au sous-shift sur lequel une opération **TF** est appliquée. Formellement, si  $\mathcal{A}$  est un alphabet,  $P \subseteq \mathcal{S}_{\mathcal{A}}^d$  un ensemble fini de motifs et  $\mathbf{T} \subseteq \mathcal{A}^{\mathbb{Z}^d}$  un sous-shift, on définit l'opération **TF** avec ensemble de motifs  $P$  par :

$$\mathbf{TF}_P(\mathbf{T}) = \mathbf{T} \setminus \left( \bigcup_{\mathbf{i} \in \mathbb{Z}^d, u \in P} \sigma^{\mathbf{i}}([u]) \right) = \mathbf{T}_{P \cup P'},$$

où l'ensemble  $P'$ , donné par la Proposition 2.2.2, est un ensemble de motifs interdits qui définit le sous-shift  $\mathbf{T}$ , c'est à dire  $\mathbf{T} = \mathbf{T}_{P'}$ .

Le sous-shift  $\mathbf{TF}_P(\mathbf{T})$  peut être vide si jamais l'ensemble  $P$  est trop restrictif. Par définition des sous-shifts de type fini, on a :

$$Cl_{\mathbf{TF}}(\mathcal{FS}) = \mathcal{SFT}.$$

**Opération produit (Prod) :** Étant donnés  $n$  sous-shifts  $\mathbf{T}_i \subseteq \mathcal{A}_i^{\mathbb{Z}^d}$  pour  $i \in \{1, \dots, n\}$ , on définit le sous-shift produit comme l'ensemble des configurations :

$$\mathbf{Prod}(\mathbf{T}_1, \dots, \mathbf{T}_n) = \{(x_1, \dots, x_n) : \forall i = 1..n, x_i \in \mathbf{T}_i\} \subseteq (\mathcal{A}_1 \times \dots \times \mathcal{A}_n)^{\mathbb{Z}^d}.$$

Ainsi, pour  $p_i \in \mathcal{A}_i^{\mathbb{S}}$ , le motif produit  $p_1 \times \dots \times p_n \in (\mathcal{A}_1 \times \dots \times \mathcal{A}_n)^{\mathbb{S}}$  est interdit dans le sous-shift  $\mathbf{Prod}(\mathbf{T}_1, \dots, \mathbf{T}_n)$  si et seulement si un des  $p_i$  est interdit dans  $\mathbf{T}_i$ .

Cette opération consiste donc à superposer les configurations des sous-shifts  $\mathbf{T}_i$ . Elle permet également de faire grossir l'alphabet d'un sous-shift. Par exemple si  $\mathbf{T}$  est un sous-shift sur l'alphabet  $\mathcal{A}$ , alors le sous-shift produit  $\mathbf{T} \times \mathcal{B}^{\mathbb{Z}^d}$  est le sous-shift inclus dans  $(\mathcal{A} \times \mathcal{B})^{\mathbb{Z}^d}$  défini par les mêmes motifs interdits que  $\mathbf{T}$ .

Le produit de deux sous-shifts de type fini est lui aussi de type fini. En terme de clôture de classe on a

$$Cl_{\mathbf{Prod}}(\mathcal{SFT}) = \mathcal{SFT} \quad Cl_{\mathbf{Prod}}(\mathcal{FS}) = \mathcal{FS}.$$

**Opération facteur (Fact) :** Cette classe d'opérations permet de transformer l'alphabet d'un sous-shift par des modifications locales, et consiste à appliquer une fonction de bloc à ce dernier (voir section 2.1.4). Soient  $\mathcal{A}$  et  $\mathcal{B}$  deux alphabets finis,  $\pi : \mathcal{A}^{\mathbb{Z}^d} \rightarrow \mathcal{B}^{\mathbb{Z}^d}$  une fonction de bloc et  $\mathbf{T} \subseteq \mathcal{A}^{\mathbb{Z}^d}$  un sous-shift. On définit  $\mathbf{Fact}_{\pi}(\mathbf{T}) := \pi(\mathbf{T})$ .

L'exemple ?? montre que la classe des sous-shifts de type fini n'est pas stable par l'opération **Fact** :

$$\mathcal{SFT} \subsetneq Cl_{\mathbf{Fact}}(\mathcal{SFT}).$$

Par définition des sous-shift sofiques, on a :

$$Cl_{\mathbf{Fact}}(\mathcal{SFT}) = \mathcal{Sofique}.$$

**Opération extension (SE) :** Cette opération permet d'augmenter la dimension d'un sous-shift. Soient  $d, d' \in \mathbb{N}^*$ , et soient  $\mathbb{G}$  et  $\mathbb{G}'$  deux sous-groupes de  $\mathbb{Z}^{d+d'}$  tels que  $\mathbb{G}$  est isomorphe à  $\mathbb{Z}^d$  et  $\mathbb{G} \oplus \mathbb{G}' = \mathbb{Z}^{d+d'}$ . Soit  $\mathbf{T} \subseteq \mathcal{A}^{\mathbb{Z}^d}$  un sous-shift de dimension  $d$ . On définit l'extension de  $\mathbf{T}$  comme le sous-shift :

$$\mathbf{SE}_{\mathbb{G}, \mathbb{G}'}(\mathbf{T}) = \left\{ x \in \mathcal{A}^{\mathbb{Z}^{d+d'}} : \forall i \in \mathbb{G}', x_{i+\mathbb{G}} \in \mathbf{T} \right\}.$$

Il est facile de vérifier que cette opération laisse stable la classe des sous-shifts de type fini  $Cl_{\mathbf{SE}}(\mathcal{SFT}) = \mathcal{SFT}$ .

### 5.1.3 La sous-action projective

La sous-action d'un sous-shift n'est pas nécessairement un sous-shift. On définit donc la sous-action projective qui consiste à ne regarder un sous-shift que selon un sous-groupe de  $\mathbb{Z}^d$ , en laissant de côté le reste du sous-shift.

**Définition.** Soit  $\mathbb{G}$  un sous-groupe de  $\mathbb{Z}^d$  librement engendré par  $u_1, u_2, \dots, u_{d'}$  ( $d' \leq d$ ). Étant donné  $\mathbf{T} \subseteq \mathcal{A}^{\mathbb{Z}^d}$  un sous-shift, on définit la *sous-action projective de  $\mathbf{T}$  sur  $\mathbb{G}$*  par :

$$\mathbf{SA}_{\mathbb{G}}(\mathbf{T}) = \left\{ y \in \mathcal{A}^{\mathbb{Z}^d} : \exists x \in \mathbf{T} \text{ tel que } \forall i_1, \dots, i_{d'} \in \mathbb{Z}^{d'}, y_{i_1, \dots, i_{d'}} = x_{i_1 u_1 + \dots + i_{d'} u_{d'}} \right\}.$$

*Exemple 5.1.1.* On se place sur  $\mathbb{Z}^2$  et on construit un sous-shift de type fini  $\mathbf{T} \subset \{0, 1, 2\}^{\mathbb{Z}^2}$  tel que la sous-action projective de  $\mathbf{T}$  sur le sous-groupe  $\Delta = \{(x, y) \in \mathbb{Z}^2 : y = x\} \subseteq \mathbb{Z}^2$  n'est pas un sous-shift de type fini. Dans cette exemple le sous-shift qui apparaît le long de  $\Delta$  est

$$\{x \in \{0, 1, 2\}^{\mathbb{Z}} : \text{les blocs de 0 consécutifs sont de longueur paire}\}.$$

L'ensemble des motifs autorisés  $\overline{F}$  est défini comme l'ensemble des motifs de taille  $4 \times 4$  suivant (le symbole  $\cdot$  peut être 1 ou 2 mais pas 0 et le symbole blanc peut être indifféremment 0, 1 ou 2) :

$$\begin{array}{|c|c|c|c|} \hline & & 2 & 0 \\ \hline & 1 & 0 & 1 \\ \hline 2 & 0 & 2 & \\ \hline 0 & 1 & & \\ \hline \end{array} ; \quad \begin{array}{|c|c|c|c|} \hline & & \cdot & \cdot \\ \hline & 2 & 0 & \cdot \\ \hline 1 & 0 & 1 & \\ \hline 0 & 2 & & \\ \hline \end{array} ; \quad \begin{array}{|c|c|c|c|} \hline & & 1 & 0 \\ \hline & 2 & 0 & 2 \\ \hline \cdot & 0 & 1 & \\ \hline \cdot & \cdot & & \\ \hline \end{array} ;$$

$$\begin{array}{|c|c|c|c|} \hline & & \cdot & \\ \hline & \cdot & \cdot & \cdot \\ \hline 2 & 0 & \cdot & \\ \hline 0 & 1 & & \\ \hline \end{array} ; \quad \begin{array}{|c|c|c|c|} \hline & & \cdot & \\ \hline & \cdot & & \cdot \\ \hline \cdot & \cdot & \cdot & \\ \hline 0 & \cdot & & \\ \hline \end{array} ; \quad \begin{array}{|c|c|c|c|} \hline & & 2 & 0 \\ \hline & \cdot & 0 & 1 \\ \hline \cdot & \cdot & \cdot & \\ \hline \cdot & & & \\ \hline \end{array} ; \quad \begin{array}{|c|c|c|c|} \hline & & \cdot & 0 \\ \hline & \cdot & \cdot & \cdot \\ \hline \cdot & & \cdot & \\ \hline \cdot & & & \\ \hline \end{array}$$

L'alternance de 1 et de 2 le long des diagonales de 0 permet de contrôler la parité des longueurs des blocs de 0 consécutifs.

Soit  $F$  l'ensemble de motifs de taille  $4 \times 4$  qui ne sont pas dans  $\overline{F}$ . Alors en appelant  $\mathbf{T}$  le sous-shift défini par l'ensemble de motifs interdits  $F$ , on a :

$$\mathbf{SA}_{\Delta}(\mathbf{T}) = \{x \in \{0, 1, 2\}^{\mathbb{Z}} : \text{les blocs de 0 consécutifs sont de longueur paire}\}$$

qui n'est pas un sous-shift de type fini (voir l'exemple ??).

*Exemple 5.1.2.* On présente ici un sous-shift de type fini  $\mathbf{T}$  tel que la sous-action  $\mathbf{SA}_{\Delta}(\mathbf{T})$  selon la diagonale  $\Delta\{(x, y) \in \mathbb{Z}^2 : y = x\}$  n'est pas sofique. En dimension 1, les sous-shifts sofiques sont exactement ceux dont le langage est régulier [LM95]. Le langage  $\{a^n b^n : n \in \mathbb{N}\}$  est l'exemple classique de langage non régulier, et c'est ce langage que l'on souhaite faire apparaître le long de  $\Delta$ .

L'alphabet du sous-shift  $\mathbf{T}$  est  $\mathcal{A} = \{*, a, b, 0, 1, 2, 3, 4\}$ , et on définit le sous-shift  $\mathbf{T} \subset \mathcal{A}^{\mathbb{Z}^2}$  de façon à obtenir, sur un fond de symboles 0, des îlots de la forme :

$$\begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & * & 1 & \dots & \dots & 1 & b & 0 \\ \hline 0 & 4 & \ddots & 1 & 1 & \ddots & 2 & 0 \\ \hline 0 & \vdots & 4 & * & b & 2 & \vdots & 0 \\ \hline 0 & \vdots & 4 & a & * & 2 & \vdots & 0 \\ \hline 0 & 4 & \ddots & 3 & 3 & \ddots & 2 & 0 \\ \hline 0 & a & 3 & \dots & \dots & 3 & * & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline \end{array} \quad (*)$$

Ainsi les seules configurations possibles sont formées d'îlots de ce type, avec éventuellement un translaté d'un (ou deux s'ils sont compatibles) des demi-plans suivants :  $2^{\mathbb{N} \times \mathbb{Z}}$ ,  $4^{-\mathbb{N} \times \mathbb{Z}}$ ,  $1^{\mathbb{Z} \times \mathbb{N}}$ ,  $3^{\mathbb{Z} \times (-\mathbb{N})}$  :

Soit  $\pi$  la 1-fonction de bloc définie localement par  $\pi(x) = 0$  pour  $x \in \{*, 0, 1, 2, 3, 4\}$  et  $\pi(a) = a$ ,  $\pi(b) = b$ . Si le sous-shift  $\mathbf{SA}_{\Delta}(\mathbf{T})$  était sofique, alors  $\pi(\mathbf{SA}_{\Delta}(\mathbf{T}))$  le serait aussi par stabilité de la classe des sofiques par fonction de bloc. Mais ce n'est pas le cas, puisque

$$\pi(\mathbf{SA}_{\Delta}(\mathbf{T})) = \mathbf{T}(\{0, a, b\}, 1, \{ba, a0, 0b\} \cup \{0a^m b^n 0 \mid m \neq n\}).$$

Ces deux exemples montrent que ni la classe des SFT ni celle des sous-shifts sofiques n'est stable par sous-action. On peut par contre montrer que la classe des sous-shifts effectifs est une classe stable par sous-action.

#### 5.1.4 La classe des sous-shifts effectifs

Un sous-shift  $\mathbf{T} \subseteq \mathcal{A}^{\mathbb{Z}^d}$  est *effectif* s'il existe un ensemble récursivement énumérable  $\mathcal{F}$  de motifs interdits, c'est à dire un ensemble  $\mathcal{F}$  énuméré par une machine de Turing tel que  $\mathbf{T} = \mathbf{T}(\mathcal{A}, d, \mathcal{F})$ . On note  $\mathcal{RE}$  l'ensemble des sous-shifts effectifs.

**Proposition 5.1.1.** *Sur  $\mathbb{Z}^d$  la classe des sous-shifts effectifs est stable par sous-action projective :*

$$Cl_{\mathbf{SA}}(\mathcal{RE}) = \mathcal{RE}.$$

*Démonstration.* Soit  $\mathbf{T} \subseteq \mathcal{A}^{\mathbb{Z}^d}$  un sous-shift effectif donné par un ensemble de motifs interdits  $\mathcal{F}$  récursivement énumérable. Soit  $\mathbb{G}$  un sous-groupe de  $\mathbb{Z}^d$  engendré par les vecteurs  $u_1, \dots, u_{d'} \in \mathbb{Z}^d$ . Le principe de la démonstration est le suivant : à partir de l'ensemble  $\mathcal{F}$  on construit un autre ensemble récursivement énumérable  $\mathcal{F}'$  tel que  $\mathbf{SA}_{\mathbb{G}}(\mathbf{T}) = \mathbf{T}_{\mathcal{F}'}$ . L'ensemble  $\mathcal{F}'$  sera le complémentaire du langage du sous-shift  $\mathbf{SA}_{\mathbb{G}}(\mathbf{T})$ . On note par  $\mathbf{T}' \subseteq \mathcal{A}^{\mathbb{Z}^{d'}}$  le sous-shift  $\mathbf{SA}_{\mathbb{G}}(\mathbf{T})$ .

Soit  $p'$  un motif dans  $\mathcal{A}^{\mathbb{S}_n^{d'}}$ . On appelle  $\Phi(p')$  l'ensemble des motifs élémentaires de  $\mathcal{A}^{\mathbb{Z}^{d'}}$  qui contiennent le motif  $p$ . Soit  $\mathcal{F}'$  l'ensemble suivant :

$$\mathcal{F}' = \bigcup_{n \in \mathbb{N}} \{p' \in \mathcal{A}^{\mathbb{S}_n^{d'}} : \forall p \in \Phi(p'), \exists \tilde{p} \in \mathcal{F} \text{ tel que } \tilde{p} \sqsubseteq p\}.$$

Il suffit de montrer que  $\mathcal{F}'$  est récursivement énumérable et que  $\mathbf{T}' = \mathbf{T}_{\mathcal{F}'}$ .

**Lemme 5.1.2.** *L'ensemble de motifs  $\mathcal{F}'$  est récursivement énumérable.*

*Démonstration du Lemme 5.1.2.* Comme  $\mathcal{F}$  est récursivement énumérable, il existe une machine de Turing  $\mathcal{M}$  dont le domaine est  $\mathcal{F}$ .

Soit  $\mathcal{M}_{\text{aux}}$  une machine de Turing auxiliaire dont le comportement sur un motif d'entrée  $p$  est le suivant : la machine  $\mathcal{M}_{\text{aux}}$  énumère tous les sous-motifs de  $p$ . Ils sont en nombre fini, et on les note  $p_1, \dots, p_n$ . Puis la machine  $\mathcal{M}_{\text{aux}}$  simule le comportement de la machine  $\mathcal{M}$  sur chacun des sous-motifs  $p_i$ , en consacrant à tour de rôle une étape de calcul à chacun des motifs. Dès que la machine  $\mathcal{M}$  s'arrête sur un des sous-motifs  $p_i$ , la machine  $\mathcal{M}_{\text{aux}}$  s'arrête aussi. Ainsi, la machine  $\mathcal{M}_{\text{aux}}$  s'arrête sur un motif  $p$  si et seulement si  $p$  possède un sous-motif dans l'ensemble  $\mathcal{F}$ .

On construit ensuite une machine de Turing  $\mathcal{M}'$  qui s'arrête sur un motif d'entrée  $p'$  si et seulement si  $p' \in \mathcal{F}'$ . Soit  $p'$  un motif dans  $\mathcal{A}^{\mathbb{S}_n^{d'}}$ . Sur le motif d'entrée  $p'$ , la machine  $\mathcal{M}'$  fonctionne de la façon suivante : elle énumère tous les supports élémentaires  $\mathbb{S}_{n+1}^{d'}, \mathbb{S}_{n+2}^{d'}, \dots$  qui contiennent le support du motif  $p'$ . On appelle cette énumération  $(\text{supp}_i)_{i \in \mathbb{N}}$ . La machine  $\mathcal{M}'$  effectue ensuite les calculs suivants pour chaque support  $\text{supp}_i$ , en consacrant à tour de rôle une étape de calcul à chaque  $\text{supp}_i$  :

- soit  $C_i$  l'ensemble fini  $\mathcal{A}^{\text{supp}_i} \cap \Phi(p')$ . On le note  $C_i = \{p_1^{(i)}, \dots, p_{k_i}^{(i)}\}$ ;
- sur chaque  $p_k^{(i)}$  à tour de rôle, la machine  $\mathcal{M}'$  simule la machine  $\mathcal{M}_{\text{aux}}$ .

De cette façon la machine  $\mathcal{M}'$  s'arrête sur un motif d'entrée  $p'$  si et seulement s'il existe un support  $\text{supp}_i$  contenant  $\text{supp}(p')$  tel que pour tout motif  $p \in \Phi(p')$  de support  $\text{supp}_i$ , le motif  $p$  contient un motif interdit de  $\mathcal{F}$ . Ceci correspond exactement à la définition de l'ensemble  $\mathcal{F}'$ , qui est donc récursivement énumérable.  $\square$

**Lemme 5.1.3.**  $\mathbf{SA}_{\mathbb{G}}(\mathbf{T}) = \mathbf{T}_{\mathcal{F}'}$ .

*Démonstration du Lemme 5.1.3.* –  $\mathbf{SA}_{\mathbb{G}}(\mathbf{T}) \subseteq \mathbf{T}_{\mathcal{F}'}$

Soit  $y \in \mathbf{SA}_{\mathbb{G}}(\mathbf{T})$ . Alors il existe une configuration  $x \in \mathbf{T}$  telle que pour toute position  $i = (i_1, \dots, i_{d'}) \in \mathbb{Z}^{d'}$ ,  $y_i = x_{i_1 u_1 + \dots + i_{d'} u_{d'}}$ . Soit  $p'$  un motif qui apparaît dans la configuration  $y$ . Si  $p'$  était dans l'ensemble  $F'$ , alors tout motif dans  $C_i$ , qui est le même ensemble que celui défini dans la démonstration du Lemme 5.1.2, contiendrait un motif interdit pour le sous-shift  $\mathbf{T}$ . En particulier, la configuration  $x$  contiendrait aussi un motif interdit pour le sous-shift  $\mathbf{T}$ , c'est-à-dire  $x \notin \mathbf{T}$  ce qui est contradictoire. Finalement la configuration  $y$  ne contient aucun motif de l'ensemble  $\mathcal{F}'$ , et donc  $y \in \mathbf{T}_{\mathcal{F}'}$ .

–  $\mathbf{T}_{\mathcal{F}'} \subseteq \mathbf{SA}_{\mathbb{G}}(\mathbf{T})$

Soit  $y \in \mathbf{T}_{\mathcal{F}'}$ . Pour tout entier  $n \in \mathbb{N}$  on a  $y_{\mathbb{S}_n^{d'}} \notin \mathcal{F}'$ , donc il existe un motif  $p_n \in \Phi(y_{\mathbb{S}_n^{d'}}) \cap \mathcal{A}^{\mathbb{S}_n^{d'}}$  qui ne contient aucun motif interdit pour le sous-shift  $\mathbf{T}$ . Par compacité, on peut construire un élément  $x \in \mathbf{T}$  tel que  $y_i = x_{i_1 u_1 + \dots + i_{d'} u_{d'}}$  pour tout  $i = (i_1, \dots, i_{d'}) \in \mathbb{Z}^{d'}$ .  $\square$

L'ensemble récursivement énumérable  $\mathcal{F}'$  est tel que  $\mathbf{T}_{\mathcal{F}'} = \mathbf{SA}_{\mathbb{G}}(\mathbf{T})$ , donc les sous-shifts effectifs sont stables par sous-action projective.  $\square$

## 5.2 Un résultat de simulation

On s'intéresse à caractériser  $Cl_{\mathbf{SA}}(\text{Sofique})$ , par la proposition 5.1.1 on a  $Cl_{\mathbf{SA}}(\text{Sofique}) \subset \mathcal{RE}$  et on va montrer que l'inclusion réciproque est vérifiée aussi. Michael Hochman montre [Hoc09] que tout sous-shift effectif de dimension  $d$  peut être obtenu comme facteur et sous-action d'un sous-shift de type fini de dimension  $d + 2$ . Sa construction amène naturellement la question suivante : est-il possible de diminuer la dimension du sous-shift de type fini à  $d + 1$ ? Une réponse positive a été apportée dans deux preuves faisant appel à des techniques de pavages différentes. L'une [DRS10] due à Bruno Durand, Andrei Romaschenko et Alexander Shen utilise une construction basée sur des pavages auto-similaires. Et celle que nous proposons [AS11], qui est une adaptation de la preuve de Robinson [Rob71] de l'indécidabilité du problème du domino, dans laquelle les zones de calcul sont remplacées par des bandes de calcul bi-infinies verticalement vues dans le chapitre précédent.

### 5.2.1 Énoncé du résultat et idée de la preuve

Le résultat dont nous allons présenter la preuve s'énonce de la manière suivante :

**Théorème 5.2.1** ([Hoc09, DRS10, AS11]). *Pour tout sous-shift effectif  $\Sigma \subset \mathcal{A}^{\mathbb{Z}^d}$  il existe un sous-shift de type fini  $\mathbf{T}(\mathcal{B}, d + 1, \mathcal{F})$  et un facteur lettre à lettre  $\pi : \mathcal{B} \rightarrow \mathcal{A}$  tel que*

$$\pi(\mathbf{T}(\mathcal{B}, d + 1, \mathcal{F})) = \widetilde{\Sigma} = \left\{ x \in \mathcal{A}^{\mathbb{Z}^{d+1}} : \exists y \in \Sigma \text{ tel que } x_{\mathbb{Z}^d \times \{i\}} = y \right\} \subset \mathcal{A}^{\mathbb{Z}^{d+1}}.$$

De plus  $h(\mathbf{T}(\mathcal{B}, d + 1, \mathcal{F})) = 0$ .

**Corollaire 5.2.2.** *Tout sous-shift effectif de dimension  $d$  peut être obtenu par facteur et sous-action d'un sous-shift de type fini de dimension  $d + 1$ , autrement dit  $Cl_{\mathbf{SA}}(\text{Sofique}) = \mathcal{RE}$ .*

**Corollaire 5.2.3.** *Tout sous-shift effectif de dimension  $d$  est conjugué à une sous-action d'un sous-shift de type fini de dimension  $d + 1$ .*

Nous allons présenter une preuve de ce résultat dans le cas particulier où  $d = 1$ , mais la construction se généralise à la dimension supérieure.

Nous commençons par donner les idées directrices de la démonstration. Soit  $\Sigma \subset \mathcal{A}_\Sigma^{\mathbb{Z}}$  un sous-shift effectif de dimension 1 sur l'alphabet  $\mathcal{A}_\Sigma$ , alors il existe une machine de Turing  $\mathcal{M}$  qui énumère les motifs interdits de  $\Sigma$ . Nous allons construire un SFT  $\mathbf{T}_{\text{Final}}$  de dimension 2. Ce sous-shift sera constitué de différents niveaux, chaque niveau étant un SFT de dimension 2, et ces niveaux seront regroupés par des opérations **Prod**, pour n'obtenir qu'un seul sous-shift au final, et des conditions de type fini **TF** de façon que chaque niveau utilise l'information contenue dans les autres niveaux. De manière plus détaillée, les différents niveaux sont :

- **Niveau 1** : ce premier niveau contient le sous-shift  $\mathcal{A}_\Sigma^{\mathbb{Z}^2}$  auquel on ajoute la condition de type fini **Align** qui impose à une même colonne de contenir un unique symbole répété verticalement. Ainsi ce niveau contient une superposition de la même configuration  $x \in \mathcal{A}_\Sigma^{\mathbb{Z}}$  (que l'on appellera candidat) ;
- **Niveau 2** : ce niveau contient les zones de calcul utilisées par  $\mathcal{M}_{\text{Forbid}}$  et  $\mathcal{M}_{\text{Search}}$ , munies d'une horloge (voir la section 4.5.4) ;
- **Niveau 3** : ce niveau contient les diagrammes espace-temps d'une machine de Turing  $\mathcal{M}_{\text{Forbid}}$  qui énumère les motifs interdits de  $\Sigma$  et vérifie qu'aucun des motifs interdits de  $\Sigma$  n'apparaît dans la configuration  $x$  ;
- **Niveau 4** : ce niveau contient les diagrammes espace-temps d'une machine de Turing  $\mathcal{M}_{\text{Search}}$  qui aide la machine  $\mathcal{M}_{\text{Forbid}}$  dans sa phase de vérification.

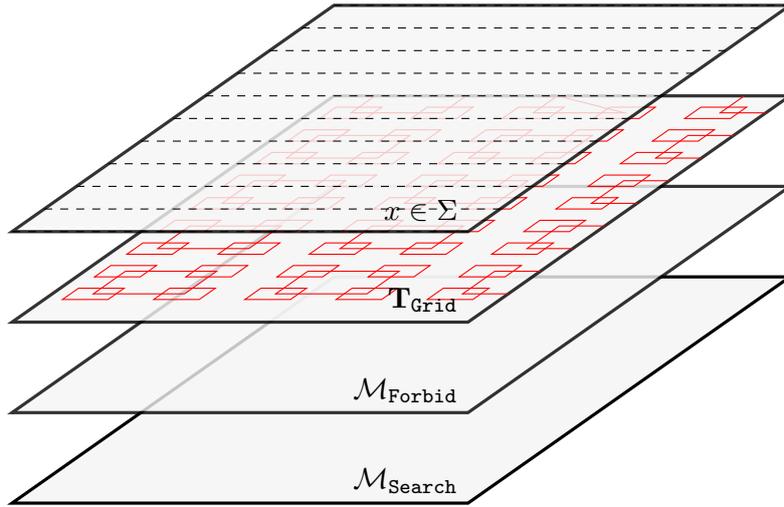


FIGURE 5.1 – Les quatres couches de la construction de  $\mathbf{T}_{\text{Final}}$ .

La difficulté est donc d'assurer qu'aucun motif interdit de  $\Sigma$  n'apparaît dans la configuration candidate  $x$ . La grille de calcul du deuxième niveau, présentée dans la section 4.5.4, offre donc la possibilité à une machine de Turing d'effectuer des calculs dans des zones de calcul finies de taille arbitrairement grande. Pour vérifier que le candidat  $x$  est bien dans  $\Sigma$ , nous utiliserons une machine de Turing  $\mathcal{M}_{\text{Forbid}}$  qui aura une double fonction : d'abord énumérer sur son ruban les motifs interdits de  $\Sigma$ , puis vérifier qu'aucun d'entre eux n'est présent dans une partie finie de la configuration  $x$ , appelée zone de responsabilité de la machine, qui dépend de la taille de la zone de calcul dans laquelle la machine évoluera. Mais le problème des espaces de calcul présentés dans la section, 4.5.2 est qu'ils sont non connexes. En conséquence, le ruban sur lequel travaille la machine  $\mathcal{M}_{\text{Forbid}}$  est un ruban fractionné, donc les motifs interdits qu'elle énumère sont eux-mêmes fractionnés. Comment alors vérifier qu'un motif interdit fractionné n'apparaît

pas dans la configuration candidate  $x$ , qui est faite de cellules connexes ?

Pour ce faire nous faisons appel à une deuxième machine de Turing  $\mathcal{M}_{\text{Search}}$ . Cette machine  $\mathcal{M}_{\text{Search}}$  reçoit des requêtes de la part de la machine  $\mathcal{M}_{\text{Forbid}}$ , sous forme d'une adresse d'une lettre de la configuration candidate  $x$  dans la zone de responsabilité de  $\mathcal{M}_{\text{Forbid}}$ . Comme toute position de la configuration  $x$  est située dans une zone de calcul d'un certain niveau, il suffit à la machine  $\mathcal{M}_{\text{Search}}$  à laquelle est formulée la requête de communiquer avec cette autre machine  $\mathcal{M}_{\text{Search}}$ . Cette communication entre machines  $\mathcal{M}_{\text{Search}}$  sera décrite dans la section 5.2.4. En supposant que la machine  $\mathcal{M}_{\text{Search}}$  remplit bien son rôle, elle répond à une requête par la lettre de la configuration  $x$  correspondante, et la machine  $\mathcal{M}_{\text{Forbid}}$  peut ainsi vérifier la validité de la configuration  $x$ . Si un motif interdit de  $\Sigma$  est détecté dans  $x$ , alors la machine  $\mathcal{M}_{\text{Forbid}}$  atteint un état spécial  $q_{\text{stop}}$ , dont la présence sera interdite dans les configurations du sous-shift final  $\mathbf{T}_{\text{Final}}$ . Cette construction, qui est résumée dans la section 5.2.5, assure que toutes les lignes  $x_{\mathbb{Z} \times \{i\}}$  dans le sous-shift final codent des configurations de  $\Sigma$ . Il reste alors à appliquer deux opérations pour obtenir  $\Sigma$  : d'abord prendre la sous-action de  $\mathbf{T}_{\text{Final}}$  sur  $\mathbb{Z} \times \{0\}$ , et ensuite effacer tous les symboles de l'alphabet de  $\mathbf{T}_{\text{Final}}$  qui ne concernent pas  $\Sigma$  (ceux utilisés pour la construction des zones de calcul, ceux codant le comportement des machines de Turing, etc...) par un facteur lettre à lettre.

Nous allons donc dans un premier temps décrire comment les différentes machines  $\mathcal{M}_{\text{Forbid}}$  et  $\mathcal{M}_{\text{Search}}$  communiquent entre elles (section 5.2.2).

## 5.2.2 Communication entre machines

### Canaux de communication

Les machines de Turing  $\mathcal{M}_{\text{Forbid}}$  et  $\mathcal{M}_{\text{Search}}$  vont avoir besoin de communiquer entre elles. Entre deux zones de calcul adjacentes de même niveau, la communication est aisée puisqu'on ne trouve que des tuiles de communication entre ces zones. Ainsi un bit d'information peut être échangé entre deux zones de calcul adjacentes de niveau  $n$  à chaque étape de calcul (voir la section 4.5.2). Mais entre deux bandes de calcul de niveaux différents le problème est plus difficile. Nous construisons dans cette partie des canaux de communication pour permettre à des machines évoluant dans des bandes de calcul de différents niveaux de communiquer entre elles. Pour construire ces canaux de communication, nous utilisons une substitution  $s_2$  sur l'alphabet  $\mathcal{G}_2$ . Cet alphabet et les règles de la substitution  $s_2$  sont décrites dans la Figure 5.2.

Nous modifions donc le sous-shift  $\mathbf{T}_{\text{Horloge}}$  défini dans la section 4.5.4 pour y ajouter ces canaux de communication. Le nouveau sous-shift est donc :

$$\mathbf{T}_{\text{Horloge}} = \mathbf{TF}_{\text{Count} \cup \text{Consist} \cup \text{Synchro}} \left( \mathbf{Prod} \left( \mathbf{T}_{s_1 \times s_2}, \mathcal{C}^{\mathbb{Z}^2} \right) \right),$$

où la substitution  $s_1 \times s_2$  est définie sur l'alphabet produit  $\mathcal{G}_1 \times \mathcal{G}_2$  et dont les règles sont formées d'une règle de  $s_1$  et d'une règle de  $s_2$ . En conséquence de quoi le sous-shift  $\mathbf{T}_{\mathcal{M}}$  est lui aussi modifié pour intégrer ces canaux de communication :

$$\mathbf{T}_{\mathcal{M}} = \mathbf{TF}_{\text{Transfert} \cup \text{Init} \cup \text{Comp} \cup \text{Bord}} \left( \mathbf{Prod} \left( \mathbf{T}_{\text{Horloge}}, \tilde{\mathcal{A}}^{\mathbb{Z}^2} \right) \right),$$

où le sous-shift  $\mathbf{T}_{\text{Horloge}}$  est celui que l'on vient de redéfinir.

Les segments rouges obtenus après itération de la substitution  $s_2$  sont appelées *lignes de communication*. Elles sont utilisées pour permettre la communication entre des bandes de calcul de différents niveaux. Les lignes de communication sont agencées de manière à former des rectangles. Les deux rectangles obtenus après  $n$  itérations de la substitution  $s_2$  sur un élément de l'alphabet  $\mathcal{G}_2$  sont appelés *rectangles de communication de niveau  $n$* . Chaque rectangle de niveau  $n$  croise deux rectangles de niveau  $n - 1$  et est intersecté par un rectangle de niveau  $n + 1$ .

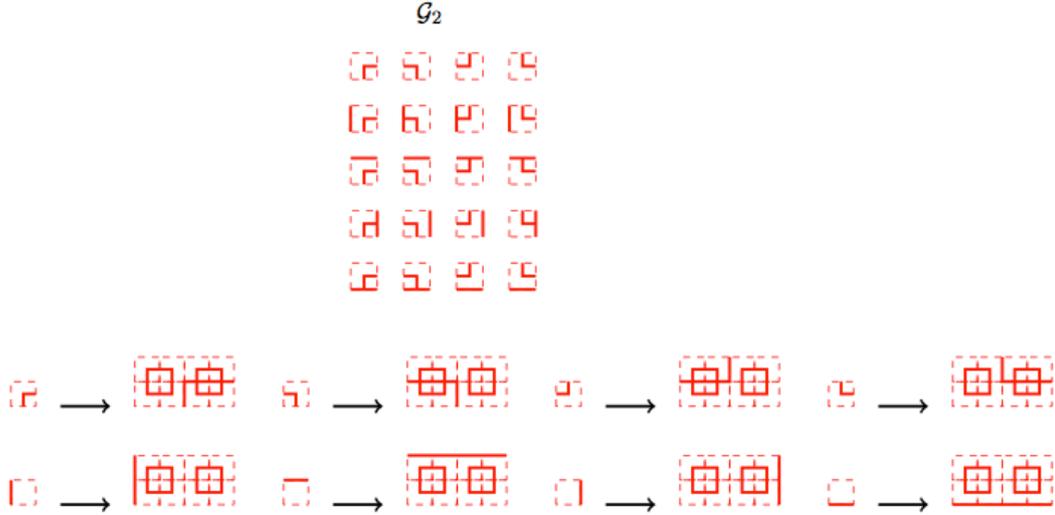


FIGURE 5.2 – L'alphabet  $\mathcal{G}_2$  et les règles de la substitution  $s_2$ .

Si l'on considère une tuile de calcul de bord  $\rightarrow$  (respectivement  $\leftarrow$ ) dans une zone de calcul de niveau  $n$ , elle est toujours située à l'intérieur d'un rectangle de communication de niveau  $n$ . Ainsi en partant de la tuile de calcul  $\rightarrow$  (respectivement  $\leftarrow$ ) et en se déplaçant vers la gauche (respectivement la droite), on rencontre nécessairement le côté gauche (respectivement le côté droit) d'un de ces rectangles. Et sur les côtés supérieur et inférieur de ce rectangle se trouvent deux tuiles de calcul  $\rightarrow$  et  $\leftarrow$ , situées dans deux zones de calcul adjacentes de niveau  $n - 1$ .

A l'aide de règles locales il est possible de construire des *canaux de communications de niveau  $n$* , qui partent de chaque tuile de calcul  $\rightarrow$  ou  $\leftarrow$  de niveau  $n$ . Un canal de communication est constitué d'une section horizontale qui rejoint un rectangle de communication de niveau  $n$  comme expliqué précédemment, et longe le bord de ce rectangle jusqu'à rencontrer une autre tuile de calcul de bord, de niveau  $n - 1$ . Ainsi une zone de calcul peut communiquer avec les quatre zones de calcul de niveau inférieur qu'elle contient (voir la Figure 5.3). Comme les zones de calcul de niveau  $n$  sont superposées verticalement avec une fréquence égale à la moitié de celle des zones de niveau  $n - 1$ , seule une moitié des zones de niveau  $n - 1$  est connecté *via* un canal de communication à une zone de niveau supérieur.

**Proposition 5.2.4.** *Pour toute zone de calcul de niveau  $n$ , il existe deux canaux de communication partant de chaque tuile de calcul  $\rightarrow$  ou  $\leftarrow$  de niveau  $n$  et terminant sur une tuile de calcul de bord de niveau  $n - 1$  (une tuile  $\rightarrow$  et une tuile  $\leftarrow$ ). Ainsi chaque zone de calcul de niveau  $n$  peut communiquer avec les quatre bandes de calcul de niveau  $n - 1$  les plus proches.*

### Adresses dans une bande de calcul

Dans cette partie nous expliquons comment définir une adresse permettant d'identifier chaque lettre  $x_i$  de la configuration candidate  $x$  située à l'intérieur d'une bande de calcul. Comme par construction du premier niveau, la configuration  $x$  est répétée verticalement, il suffit de définir une adresse qui identifie la  $i^{\text{ème}}$  colonne dans le sous-shift de type fini du premier niveau.

Soit  $C_n$  une zone de calcul de niveau  $n$  d'une configuration  $x \in \mathbf{T}_{\text{Grid}}$  et soit  $S_n$  la bande de calcul associée. Par la Proposition 4.5.2, il existe un unique  $i \in [0, 4^n - 1] \times [0, 2^n - 1]$  et un unique  $y \in \mathbf{T}_{\text{Grid}}$  tels que  $s_{\text{Grid}}^n(y) = \sigma^i(x)$ . Donc il existe un unique  $(j_1, j_2) \in \mathbb{Z}^2$  tel que

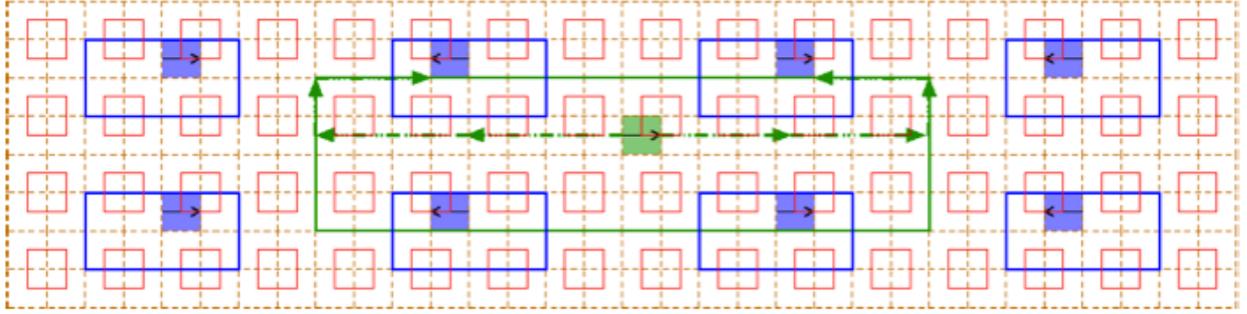


FIGURE 5.3 – Les rectangles de communication permettent à chaque zone de calcul de niveau  $n$  de communiquer avec les quatre zones de calcul de niveau  $n - 1$  les plus proches.

$C_n \sqsubset s_{\text{Grid}}^n(y_{\{j_1, j_2\}})$ . On a  $S_n \sqsubset \sigma^{-i}(s_{\text{Grid}}^n(y_{\{j_1\} \times \mathbb{Z}})) \sqsubset x$ , et la bande  $\sigma^{-i}(s_{\text{Grid}}^n(y_{\{j_1\} \times \mathbb{Z}}))$  est appelée *bande de dépendance* associée à la bande de calcul  $S_n$ .

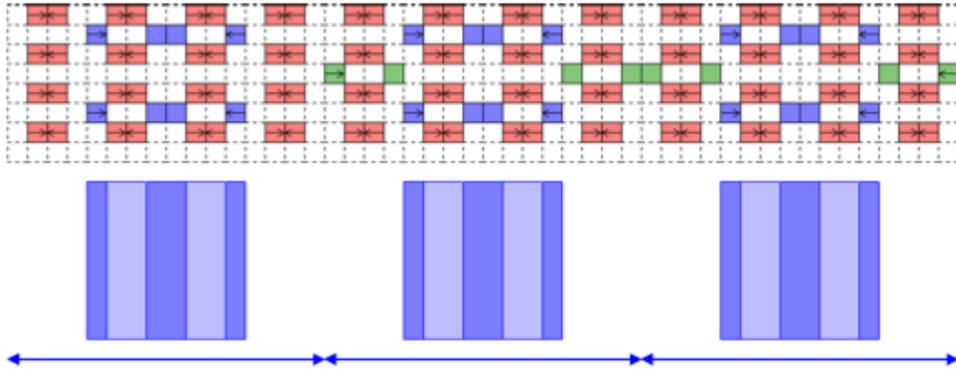


FIGURE 5.4 – Les bandes de dépendance associées à des zones de calcul de niveau 2.

Dans le sous-shift  $\mathbf{T}_{\text{Grid}}$ , le ruban d'une machine de Turing dans une bande de calcul de niveau  $n$  est fractionné. De ce fait, une machine de Turing de niveau  $n$  ne peut pas accéder directement à toutes les colonnes qui composent sa bande de dépendance. Pour obtenir ces informations, la machine communique avec une machine de Turing de niveau inférieur (voir la Partie 5.2.4) mais il faut pour cela que les deux machines puissent identifier précisément de quelle colonne il est question.

Étant donnée une bande de dépendance associée à une bande de calcul de niveau  $n$ , il est possible d'explicitier des coordonnées relatives à cette bande pour chaque colonne. Ces adresses sont composées de  $n$  symboles choisis parmi l'alphabet à quatre éléments  $\{0, 1, 2, 3\}$ . Chacun des motifs  $s_1^n(a)$  peut être décomposé horizontalement en quatre (éventuellement différents) motifs  $s_1^{n-1}(b)$  où  $a, b$  sont des lettres de l'alphabet  $\mathcal{G}_1$ . Le premier symbole de l'adresse indique à laquelle des bandes de dépendances de niveau  $n - 1$  la colonne appartient. En itérant ce procédé, la position d'une colonne dans une bande de dépendance de niveau  $n$  est donnée par un mot de  $n$  lettres sur l'alphabet  $\{0, 1, 2, 3\}$ , que l'on appelle *adresse de la colonne* (voir la Figure 5.5).

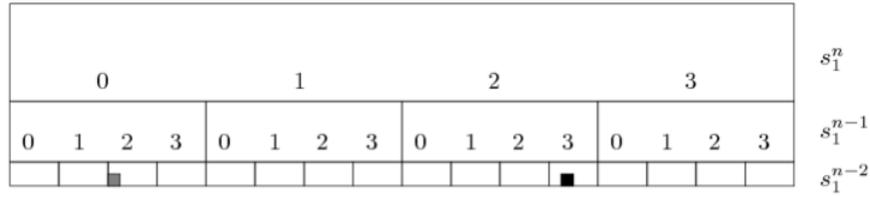


FIGURE 5.5 – Adresses de deux colonnes dans une bande de dépendance de niveau 3. L'adresse de la colonne repérée par une tuile noire est 231, tandis que l'adresse de la colonne repérée par une tuile grise est 020.

**Proposition 5.2.5.** *Pour chaque zone de dépendance de niveau  $n$ , il est possible de décrire la position de chaque colonne de la bande par une adresse de  $n$  bits sur un alphabet à quatre éléments.*

### Zones de responsabilité

Rappelons que la machine de Turing  $\mathcal{M}_{\text{Forbid}}$ , qui sera décrite plus précisément dans la Partie 5.2.4, a un double rôle : elle énumère les motifs interdits de  $\Sigma$  et vérifie que ces motifs n'apparaissent pas dans la configuration  $x$ . Mais vérifier l'absence d'un motif interdit dans l'intégralité de la configuration  $x$  nécessite une infinité d'étapes de calcul, c'est pourquoi chacune des machines  $\mathcal{M}_{\text{Forbid}}$  ne se charge de la vérification que d'une partie finie de la configuration  $x$ . Cette partie finie est appelée *zone de responsabilité* de la machine  $\mathcal{M}_{\text{Forbid}}$ .

Nous assignons donc une zone de responsabilité à chaque bande de calcul. Pour une bande de calcul de niveau  $n$  cette zone de responsabilité est formée de  $3 * (2 * 4^{n-1}) = 6 * 4^{n-1}$  cellules, centrées autour de la bande de calcul (voir la Figure 5.6), de sorte que la zone de responsabilité d'une machine de niveau  $n$  débute là où la zone de responsabilité de sa voisine de gauche de même niveau termine.

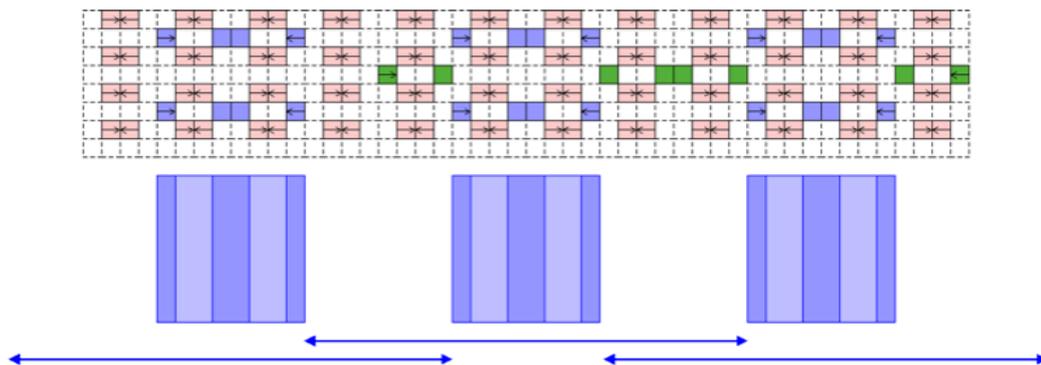


FIGURE 5.6 – Zones de responsabilité pour des bandes de niveau 2. Ces zones sont formées de 24 cellules et se chevauchent sur 8 cellules.

Ainsi définies, les zones de responsabilité se chevauchent : deux zones de responsabilité adjacentes de même niveau  $n$  partagent  $2 * 4^{n-1}$  cellules. Ces chevauchements sont essentiels : sans eux, on pourrait imaginer un motif interdit à cheval sur deux zones de responsabilité, et qui ne serait ainsi jamais détecté. Pour la même raison il est important que la taille des chevauchements croisse avec le niveau des machines, cela nous assure que tout motif de la

configuration candidate  $x$  est contenu dans une infinité de zones de responsabilité de niveaux croissants.

### 5.2.3 Génération de motifs interdits

Rappelons que comme le sous-shift  $\Sigma$  est récursivement énumérable, il existe une machine de Turing qui énumère ses motifs interdits. Nous décrivons dans cette partie une version modifiée de cette machine, qui en plus de l'énumération vérifie que chaque motif interdit qu'elle produit n'apparaît pas dans sa zone de responsabilité, les comparaisons se faisant avec les symboles du sous-shift du premier niveau. Les zones de calcul sont fractionnées (voir la Figure 4.9), donc durant un calcul sur une bande de calcul de taille  $2^n$ , une machine  $\mathcal{M}_{\text{Forbid}}$  ne peut pas accéder entièrement à sa zone de responsabilité. La machine  $\mathcal{M}_{\text{Forbid}}$  sollicitera l'aide d'une deuxième machine de Turing  $\mathcal{M}_{\text{Search}}$  afin d'obtenir les symboles de la configuration candidate  $x \in \mathcal{A}_\Sigma^{\mathbb{Z}}$  situés à l'intérieur de sa zone de responsabilité, et dont elle aura besoin. Le comportement de la machine  $\mathcal{M}_{\text{Forbid}}$  est le suivant : elle énumère autant de motifs interdits de  $\Sigma$  que la taille de sa zone de calcul le lui permet, et chaque fois qu'un motif est produit, la machine  $\mathcal{M}_{\text{Forbid}}$  vérifie que ce motif n'apparaît pas dans sa zone de responsabilité. Nous allons décrire ce fonctionnement plus précisément, en commençant par présenter les trois rubans dont dispose la machine  $\mathcal{M}_{\text{Forbid}}$ .

**Rubans de la machine  $\mathcal{M}_{\text{Forbid}}$**  La machine  $\mathcal{M}_{\text{Forbid}}$  utilise trois rubans :

- le premier ruban est le *ruban de calcul* ;
- le deuxième ruban est le *ruban d'écriture*, sur lequel les motifs interdits sont écrits les uns après les autres ;
- le troisième et dernier ruban est le *ruban de communication*, qui contient les adresses (une adresse remplaçant la précédente) des lettres de l'alphabet  $\mathcal{A}_\Sigma$  dont la machine  $\mathcal{M}_{\text{Forbid}}$  a besoin pour vérifier que le motif écrit sur son ruban d'écriture n'apparaît pas dans sa zone de responsabilité. La machine  $\mathcal{M}_{\text{Forbid}}$  attend que la machine  $\mathcal{M}_{\text{Search}}$ , choisie parmi les trois machines  $\mathcal{M}_{\text{Search}}$  de son voisinage (à sa gauche, au centre ou à sa droite), qu'elle a sollicitée soit disponible. Puis lui transfère l'adresse de la lettre de  $\mathcal{A}_\Sigma$  dont elle a besoin (voir la section 5.2.4).

Nous développons à présent les différentes étapes d'un calcul de la machine  $\mathcal{M}_{\text{Forbid}}$ .

**Calcul de la taille de la zone de responsabilité** La machine  $\mathcal{M}_{\text{Forbid}}$  commence par calculer la taille de sa zone de calcul, comprise entre les tuiles de calcul  $\boxrightarrow$  et  $\boxleftarrow$ . De cette façon, la machine  $\mathcal{M}_{\text{Forbid}}$  connaît la taille de sa zone de responsabilité, et ceci peut être effectué en temps linéaire par rapport à la taille de la zone de calcul.

**Énumération de motifs interdits** Une fois le calcul précédent effectué, la machine  $\mathcal{M}_{\text{Forbid}}$  énumère les motifs interdits de  $\Sigma$ , et dès qu'un motif est intégralement inscrit sur son ruban d'écriture, elle vérifie que ce motif n'apparaît pas à l'intérieur de sa zone de responsabilité.

**Vérification de la zone de responsabilité** Supposons que la machine  $\mathcal{M}_{\text{Forbid}}$  a inscrit sur son ruban d'écriture le motif interdit  $f = f_0 f_1 \dots f_{k-1}$ , et que la machine  $\mathcal{M}_{\text{Forbid}}$  est chargée d'une zone de responsabilité de niveau  $n$  que l'on appellera  $a_0 a_1 \dots a_{6*4^{n-1}-1}$ . Elle commence par demander à une des machines  $\mathcal{M}_{\text{Search}}$  voisines la lettre  $a_0$  dont l'adresse est  $0 \dots 0$  où le symbole 0 est répété  $n$  fois (le principe d'une telle requête sera développé dans la Partie 5.2.4), et compare la lettre  $a_0$  avec la première lettre du motif interdit  $f_0$ . Si les lettres sont identiques, il est encore possible que le motif interdit  $f$  apparaisse en position 0 dans la zone de responsabilité,

donc la machine garde en mémoire la fait qu'il faille continuer la comparaison des motifs  $f$  et  $a_0 \dots a_k$ . Si  $f_0 \neq a_0$  on est sûr que le motif interdit  $f$  n'apparaît pas à cette position. Dans les deux cas, la machine  $\mathcal{M}_{\text{Forbid}}$  fait à nouveau appel à une machine  $\mathcal{M}_{\text{Search}}$  pour obtenir le deuxième symbole de sa zone de responsabilité  $a_1$  d'adresse  $0 \dots 01$  où le symbole  $0$  est répété  $n - 1$  fois. Si cela est pertinent, elle compare  $a_1$  avec  $f_1$ . Puis elle compare  $a_1$  avec  $f_0$ , et ainsi de suite. Si à un moment la machine  $\mathcal{M}_{\text{Forbid}}$  détecte que  $f = a_i \dots a_{i+k+1}$  pour un certain  $i$ , elle arrête tout calcul et entre dans un état  $q_{\text{stop}}$  dans lequel elle restera jusqu'à la réinitialisation de l'horloge. Cet état  $q_{\text{stop}}$  sera interdit dans le sous-shift final. Dans le pire des cas (si aucun motif interdit n'est détecté) pour chaque motif interdit  $f$  de longueur  $k$ , la machine  $\mathcal{M}_{\text{Forbid}}$  a besoin de  $6 * 4^{n-1} * k * t(n)$  étapes de calcul, où  $t(n)$  est le temps nécessaire à la machine  $\mathcal{M}_{\text{Search}}$  pour répondre à une requête de  $\mathcal{M}_{\text{Forbid}}$ . Dans la Partie 5.2.4 nous donnerons une estimation de ce temps de réponse.

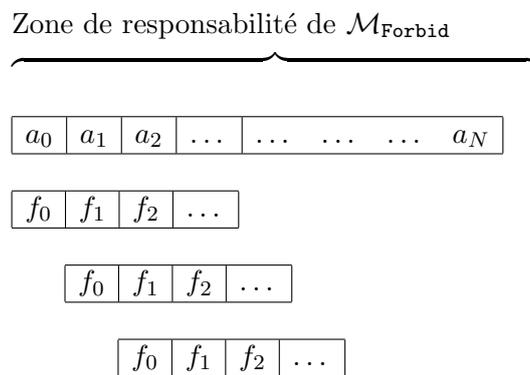


FIGURE 5.7 – Lorsqu'un motif interdit de  $\Sigma f = f_0 f_1 \dots f_k$  est énuméré par la machine  $\mathcal{M}_{\text{Forbid}}$ , la recherche de ce motif à toutes les positions possibles de la zone de responsabilité de  $\mathcal{M}_{\text{Forbid}}$  est effectuée en parallèle.

### 5.2.4 Détection de motifs interdits

La machine de Turing  $\mathcal{M}_{\text{Search}}$  reçoit une *requête*, c'est-à-dire une suite de symboles codant l'adresse d'une lettre à l'intérieur d'une zone de responsabilité d'une machine  $\mathcal{M}_{\text{Forbid}}$ , de la part d'une machine  $\mathcal{M}_{\text{Forbid}}$  chaque fois qu'une adresse est intégralement inscrite sur le ruban de communication de la machine  $\mathcal{M}_{\text{Forbid}}$ . La machine  $\mathcal{M}_{\text{Search}}$  doit renvoyer la lettre correspondante sur le premier niveau de construction  $\mathcal{A}_{\Sigma}^{\mathbb{Z}^2}$ . Il faut remarquer que la zone de responsabilité d'une machine  $\mathcal{M}_{\text{Forbid}}$  de niveau  $n$  ne correspond pas tout à fait à l'ensemble des positions qu'une machine  $\mathcal{M}_{\text{Search}}$  de même niveau peut atteindre en faisant appel aux machines d'ordre inférieur. En fait une machine  $\mathcal{M}_{\text{Forbid}}$  partage sa zone de responsabilité avec trois machines  $\mathcal{M}_{\text{Search}}$ , et selon l'adresse du symbole recherché, la machine  $\mathcal{M}_{\text{Forbid}}$  envoie sa requête à la machine  $\mathcal{M}_{\text{Search}}$  idoïne (voir la Figure 5.8 pour un exemple).

Décrivons plus précisément le comportement d'une machine  $\mathcal{M}_{\text{Search}}$ , en commençant par détailler les cinq rubans qu'elle utilise.

**Rubans d'une machine  $\mathcal{M}_{\text{Search}}$**  Une machine  $\mathcal{M}_{\text{Search}}$  de niveau  $n$  utilise cinq rubans :

- le premier ruban est le *ruban de calcul* ;
- le deuxième ruban est le *ruban de requête d'ordre supérieur*, c'est sur ce ruban que les bits d'une adresse transférée par une machine  $\mathcal{M}_{\text{Search}}$  de niveau  $n + 1$  sont écrits.

- les trois derniers rubans sont le *ruban de requête gauche*, le *ruban de requête central* et le *ruban de requête droite* sur lesquels seront inscrits les adresses des symboles requis par la machine  $\mathcal{M}_{\text{Forbid}}$  de niveau  $n$  située respectivement à la gauche, au même endroit et à la droite de la bande de calcul de la machine  $\mathcal{M}_{\text{Search}}$  considérée.

**Requête envoyée par une machine  $\mathcal{M}_{\text{Forbid}}$**  A chaque fois qu’une adresse est écrite sur le ruban de communication d’une machine  $\mathcal{M}_{\text{Forbid}}$ , celle-ci envoie cette requête à la machine  $\mathcal{M}_{\text{Search}}$  de même niveau située soit dans la même bande de calcul, soit dans la bande de calcul directement à gauche ou directement à droite. La machine  $\mathcal{M}_{\text{Forbid}}$  envoie les bits formant l’adresse un par un (un bit est envoyé à chaque étape de calcul), et donc une machine de niveau  $n$  envoie un bit toutes les  $2^n$  lignes (avec l’implémentation des machines de Turing présentée dans la section 4.2). Deux bandes de calcul adjacentes de même niveau peuvent communiquer grâce aux canaux de communication décrits dans la Partie 5.2.2, en utilisant le fait qu’une ligne ne comprend que des zones de calcul de même niveau (voir la Proposition 4.5.3). Les bits de l’adresse étant envoyés un à un, le transfert de la totalité de l’adresse nécessite  $2^n * n$  lignes. La requête est alors écrite sur le ruban de requête correspondant. La machine  $\mathcal{M}_{\text{Forbid}}$  attend ensuite la réponse de la machine  $\mathcal{M}_{\text{Search}}$  correspondante avant de poursuivre son calcul.

**Requête envoyée par une machine  $\mathcal{M}_{\text{Search}}$**  Une machine  $\mathcal{M}_{\text{Search}}$  de niveau  $n \geq 2$  peut envoyer sa requête à l’une des quatre machines  $\mathcal{M}_{\text{Search}}$  de niveau  $n - 1$  situées dans sa zone de dépendance. De manière similaire à ce qui se passe pour une machine  $\mathcal{M}_{\text{Forbid}}$ , l’envoi des  $n$  bits composant l’adresse se fait bit par bit et nécessite donc  $2^n * n$  lignes de calcul. La machine  $\mathcal{M}_{\text{Search}}$  de niveau  $n$  utilise elle aussi les canaux de communication décrits dans la Proposition 5.2.4 pour communiquer : chaque tuile de calcul de bord  $\Rightarrow$  ou  $\Leftarrow$  est située à l’intérieur d’un rectangle de niveau  $n$  qui leur permet de communiquer avec une tuile de calcul de bord d’une zone de calcul de niveau  $n - 1$ .

**Traitement d’une requête : l’arbre de recherche** Une machine  $\mathcal{M}_{\text{Search}}$  de niveau  $n$  répond aux requêtes écrites sur ses différents rubans, en consacrant une étape de calcul sur quatre à chacun de ses rubans. L’adresse inscrite sur le ruban est recopiée sur le ruban de calcul, et la machine garde en mémoire de quel ruban provient la requête qu’elle est en train de traiter. Si la machine  $\mathcal{M}_{\text{Search}}$  est une machine de niveau 1, elle peut directement lire la lettre de  $\mathcal{A}_\Sigma$ . Sinon elle est de niveau  $n$ , et transmet l’adresse à la machine  $\mathcal{M}_{\text{Search}}$  de niveau  $n - 1$  : le premier bit de l’adresse indique quel canal de communication utiliser, et seuls les  $n - 1$  derniers bits sont transmis. Puis la machine  $\mathcal{M}_{\text{Search}}$  de niveau  $n$  attend la réponse à cette requête. Dès qu’une machine de niveau 1 reçoit la requête, le symbole correspondant dans  $\mathcal{A}_\Sigma$  est trouvé (voir la Figure 5.8). Il ne reste plus qu’à remonter l’arbre de recherche pour faire parvenir le symbole à la machine  $\mathcal{M}_{\text{Forbid}}$  qui avait initialement formulé la requête.

**Réponse à une requête : remonter l’arbre de recherche** Lorsqu’une machine  $\mathcal{M}_{\text{Search}}$  reçoit de la machine de niveau inférieur le symbole correspondant à la requête qu’elle lui avait formulée, elle le transfère par le même canal de communication que celui d’où lui est venu cette requête. Cette opération se fait en une seule étape de calcul pour deux raisons. D’abord parce qu’il n’y a qu’un seul symbole à transmettre, mais aussi parce que la machine à laquelle ce symbole est envoyé attend cette réponse et donc sait de quel symbole il s’agit. Une machine  $\mathcal{M}_{\text{Search}}$  finit toujours par répondre à une requête d’une machine  $\mathcal{M}_{\text{Forbid}}$  partageant la même bande de calcul, puisque chacune des  $\mathcal{M}_{\text{Search}}$  travaille à tour de rôle pour les machines  $\mathcal{M}_{\text{Forbid}}$  de même niveau et la machine  $\mathcal{M}_{\text{Search}}$  de niveau supérieur.

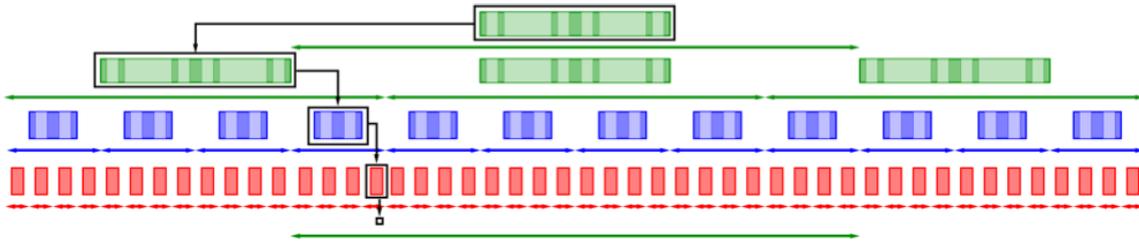


FIGURE 5.8 – Un exemple d'arbre de recherche correspondant à une requête par une machine  $\mathcal{M}_{\text{Forbid}}$  de niveau 3. Selon l'adresse de la lettre requise, la machine  $\mathcal{M}_{\text{Forbid}}$  envoie la requête à une des trois machines  $\mathcal{M}_{\text{Search}}$ .

**Initialisation des calculs** Lorsque l'horloge est réinitialisée, il est crucial de ne pas effacer les adresses écrites sur les rubans de requêtes des machines  $\mathcal{M}_{\text{Search}}$ , car des machines de niveau supérieur sont potentiellement en train d'attendre une réponse.

Un autre problème pouvant survenir lors d'une réinitialisation est le suivante : une machine a envoyé une requête, et elle est réinitialisée alors qu'elle attend toujours la réponse à sa requête. Pour parer à cette difficulté on impose que la première chose que fasse une machine lorsqu'elle est réinitialisée est d'attendre la réponse à une éventuelle requête laissée en suspens dans son calcul précédent, et d'ignorer cette réponse avant de recommencer un nouveau calcul.

**Temps de réponse d'une machine  $\mathcal{M}_{\text{Search}}$**  On appelle  $t(n)$  le temps nécessaire à une machine  $\mathcal{M}_{\text{Search}}$  de niveau  $n$  pour répondre à une requête d'une machine  $\mathcal{M}_{\text{Forbid}}$  de même niveau. Comme une machine de niveau  $n$  effectue une étape de calcul toutes les  $2^n$  lignes, une machine  $\mathcal{M}_{\text{Search}}$  de niveau  $n$  a besoin de  $2^n * t(n)$  lignes pour répondre à une requête d'une machine  $\mathcal{M}_{\text{Forbid}}$ .

Une machine  $\mathcal{M}_{\text{Search}}$  de niveau  $n \geq 2$  est aidée par une machine  $\mathcal{M}_{\text{Search}}$  de niveau  $n - 1$  : elle lui transfère un à un les  $n - 1$  derniers bits de l'adresse de la requête (pour cela elle a besoin de  $n * 2^n$  lignes). Puis elle attend la réponse de la machine  $\mathcal{M}_{\text{Search}}$  de niveau  $n - 1$ . Il est possible que cette machine  $\mathcal{M}_{\text{Search}}$  de niveau  $n - 1$  soit déjà en train de répondre à d'autres requêtes, dans le pire des cas trois autres requêtes. Au total une machine  $\mathcal{M}_{\text{Search}}$  de niveau  $n - 1$  peut être amenée à répondre à quatre requêtes simultanément. Au final, le temps de réponse d'une machine  $\mathcal{M}_{\text{Search}}$  de niveau  $n$  est donné par la formule de récurrence :

$$2^n t(n) \leq n * 2^n + 4 * 2^{n-1} * t(n - 1).$$

De l'inégalité précédente nous déduisons que

$$t(n) \leq 2^n * \mathcal{O}(n^2 2^n).$$

Ainsi une machine  $\mathcal{M}_{\text{Search}}$  de niveau  $n$  répond à une requête d'une machine  $\mathcal{M}_{\text{Forbid}}$  en  $\mathcal{O}(n^2 2^n)$  étapes de calcul.

**Proposition 5.2.6.** *Toutes les requêtes d'une machine  $\mathcal{M}_{\text{Forbid}}$  de niveau  $n$  reçoivent une réponse d'une machine  $\mathcal{M}_{\text{Search}}$  de même niveau en au plus  $n^2 2^n$  étapes de calcul, pour un  $n$  assez grand.*

**Temps de vérification par  $\mathcal{M}_{\text{Forbid}}$**  Supposons qu'une machine  $\mathcal{M}_{\text{Forbid}}$  ait énuméré sur son ruban un motif  $f$  de taille  $k$ . Elle doit à présent s'assurer que ce motif n'apparaît pas dans sa zone de responsabilité. D'après ce qui a été présenté dans la Partie 5.2.4, ceci nécessite  $6 * 4^{n-1} * k * t(n) \leq k * n^2 * 2^{3n+1}$  étapes de calcul.

Soit  $(f_i)_{i \in \mathbb{N}}$  l'énumération des motifs interdits de  $\Sigma$  faite par la machine  $\mathcal{M}_{\text{Forbid}}$ . On appelle  $t(f_0, \dots, f_k)$  le temps nécessaire à la machine  $\mathcal{M}_{\text{Forbid}}$  pour vérifier que les motifs  $(f_i)_{i \in [0, k]}$  n'apparaissent pas dans sa zone de responsabilité, et  $t'(f_0, \dots, f_k)$  le temps nécessaire à cette même machine pour uniquement énumérer ces motifs sans vérifier leur absence dans sa zone de responsabilité. Le temps nécessaire à une machine  $\mathcal{M}_{\text{Forbid}}$  de niveau  $n$  pour vérifier l'absence des motifs interdits  $(f_i)_{i \in [0, k]}$  est donc

$$t(f_0, \dots, f_k) \leq t'(f_0, \dots, f_k) + (k + 1) * \max\{|f_i| : i \in [0, k]\} * n^2 * 2^{3n+1}.$$

Comme  $t'(f_0, \dots, f_k)$  ne dépend pas du niveau de la machine  $\mathcal{M}_{\text{Forbid}}$ , et qu'une machine  $\mathcal{M}_{\text{Forbid}}$  de niveau  $n$  peut effectuer  $2^{2^n} + 2$  étapes de calcul (voir le Théorème 4.5.5), il existe un niveau  $n$  tel que toutes les machines de ce niveau vérifient que les motifs interdits  $(f_i)_{i \in [0, k]}$  n'apparaissent pas dans leurs zones de responsabilité.

**Proposition 5.2.7.** *Pour tout motif interdit de  $\Sigma$ , il existe un entier  $n \in \mathbb{N}$  tel que dans tout calcul dans une bande de niveau  $n$ , le mot  $w$  est énuméré par la machine  $\mathcal{M}_{\text{Forbid}}$  et celle-ci a le temps de vérifier qu'il n'apparaît pas dans sa zone de responsabilité.*

### 5.2.5 Construction finale

Nous résumons la construction finale du sous-shift  $\mathbf{T}_{\text{Final}}$  étape par étape.

1. Tout d'abord nous construisons les quatre niveaux du sous-shift :

$$\mathbf{T}_{\text{Level}} = \mathbf{Prod} \left( \mathcal{A}^{\mathbb{Z}^2}, \mathbf{T}_{\text{Grid}}, \mathcal{A}_{\text{Comp}(\mathcal{M}_{\text{Forbid}})}^{\mathbb{Z}^2}, \mathcal{A}_{\text{Comp}(\mathcal{M}_{\text{Search}})}^{\mathbb{Z}^2} \right).$$

2. Ensuite, nous alignons verticalement les lettres du premier niveau pour obtenir la même configuration candidate sur chaque ligne :

$$\mathbf{T}_{\text{Align}} = \mathbf{TF}_{\text{Align}}(\mathbf{T}_{\text{Level}}).$$

3. Puis nous insérons les calculs des machines  $\mathcal{M}_{\text{Forbid}}$  et  $\mathcal{M}_{\text{Search}}$  à l'aide des conditions de type fini  $\mathbf{Work}_{\mathcal{M}_{\text{Forbid}}} \cup \mathbf{Work}_{\mathcal{M}_{\text{Search}}}$ , et nous y ajoutons les communications entre les différents niveaux de la construction, toujours avec des conditions de type fini  $\mathbf{Com}$ . Enfin nous finissons par la condition  $\mathbf{Forbid}$  qui permet d'éliminer les configurations dans lesquelles une machine  $\mathcal{M}_{\text{Forbid}}$  détecte un motif interdit. Nous obtenons alors :

$$\mathbf{T}_{\text{Final}} = \mathbf{TF}_{\mathbf{Work}_{\mathcal{M}_{\text{Forbid}}} \cup \mathbf{Work}_{\mathcal{M}_{\text{Search}}} \cup \mathbf{Com} \cup \mathbf{Forbid}}(\mathbf{T}_{\text{Align}}).$$

Appelons  $\mathbf{T}$  le sous-shift  $\mathbf{Fact}_{\pi}(\mathbf{SA}_{(1,0)\mathbb{Z}}(\mathbf{T}_{\text{Final}}))$ , où  $\pi$  est le facteur qui consiste à ne conserver que les lettres de l'alphabet  $\mathcal{A}_{\Sigma}$  du premier niveau. Montrons que les sous-shifts  $\Sigma$  et  $\mathbf{T}$  sont en fait identiques.

*Démonstration.* –  $\Sigma \subseteq \mathbf{T}$  : Soit  $x \in \Sigma$ , par construction du sous-shift  $\mathbf{T}_{\text{Final}}$  il est facile de construire une configuration bidimensionnelle  $y$  telle que  $y \in \mathbf{T}_{\text{Final}}$  et  $\pi(y|_{(1,0)\mathbb{Z}}) = x$ .  
–  $\mathbf{T} \subseteq \Sigma$  : Soit  $x \in \mathbf{T}$ , nous montrons que  $x \in \Sigma$ . Par définition, il existe  $y \in \mathbf{T}_{\text{Final}}$  tel que  $\pi(y|_{\mathbb{Z}e_1}) = x$ . Il nous suffit de montrer que tout motif qui apparaît dans  $x$  est dans le langage  $\mathbf{L}(\Sigma)$ . Soit  $w$  un motif qui apparaît dans  $x$ . Supposons que  $w$  n'est pas dans  $\mathbf{L}(\Sigma)$ .

Alors par la Proposition 5.2.7, il existe un  $n \in \mathbb{N}$  tel que dans tout calcul dans une bande de niveau  $n$ , le mot  $w$  est énuméré par la machine  $\mathcal{M}_{\text{Forbid}}$  et celle-ci a eu le temps de vérifier qu'il n'apparaît pas dans sa zone de responsabilité. En particulier le motif  $w$  a été comparé avec tous les autres motifs de même longueur qui apparaissent dans  $x$ . Comme  $w$  apparaît dans  $x$ , cela signifie qu'il existerait une bande de calcul de niveau  $n$  dans laquelle la condition de type finie **Forbid** n'est pas respectée, d'où la contradiction.  $\square$

### 5.3 Autres travaux

Dans [PS10], les auteurs étudient la classe  $\mathcal{Cl}_{\text{SA}}(\mathcal{SFT})$  et montrent qu'elle contient tous les sous-shifts sofiques d'entropie strictement positive. Cependant la classe  $\mathcal{Cl}_{\text{SA}}(\mathcal{SFT})$  n'est pas complètement caractérisée, et savoir quels sont les sous-shifts qui peuvent être obtenus par sous-action d'un sous-shift de type fini reste une question ouverte.

De manière plus générale, on s'aperçoit qu'il est facile d'obtenir un sous-shift effectif unidimensionnel d'entropie positive comme sous-action projective d'un SFT si ce dernier contient un sous-shift sofique d'entropie positive [SS12] l'enjeu étant plutôt de réaliser des sous-shift effectif d'entropie nulle.

## Chapitre 6

# Quelques applications du théorème précédent

Le théorème 5.2.1 a un rôle central lorsque on cherche à construire un SFT ou un espace de pavage par règles locales avec des propriétés particulières. Voici quelques exemples :

1. caractérisation de l'entropie d'un SFT multidimensionnel [HM10],
2. caractérisation des pavages par coupé projection qui admettent des règles locales [FS12],
3. caractérisation des pavages s-adique multidimensionnel qui admettent des règles locales,
4. caractérisation des périodes possible pour un SFT multidimensionnel,
5. calculabilité de la fonction de quasi-périodicité...

### 6.1 Entropie atteignable par un sous-shift de type fini ou un sofique

En dimension 1, pour étudier l'entropie d'un SFT, il suffit de chercher le rayon spectral de la matrice d'adjacence du graphe des mots. On a alors une caractérisation des entropies possibles.

**Théorème 6.1.1** ([Lin84]). *On a :*

$$\left\{ h(\mathbf{T}) : \mathbf{T} \subset \mathcal{A}^{\mathbb{Z}^d} \text{ SFT} \right\} = \left\{ \frac{p}{q} \log(\alpha) : p, q \in \mathbb{N}, \alpha \text{ Nombre de Perron} \right\}$$

*où un nombre de Perron est un nombre algébrique plus grand que le module de ses conjugués.*

En dimension supérieure la caractérisation des entropies d'un sous-shift de type fini et d'un sofique ont été démontré récemment en utilisant des techniques d'informatique théorique.

**Théorème 6.1.2** ([HM10]). *Pour  $d \geq 2$ , on a*

$$\left\{ h(\mathbf{T}) : \mathbf{T} \subset \mathcal{A}^{\mathbb{Z}^d} \text{ SFT} \right\} = \left\{ h(\mathbf{T}) : \mathbf{T} \subset \mathcal{A}^{\mathbb{Z}^d} \text{ sofique} \right\} = \left\{ \alpha : \exists (\alpha_n) \in \mathbb{Q}^{\mathbb{N}} \searrow \text{réursive telle que } \alpha = \inf_{n \in \mathbb{N}} \alpha_n \right\}.$$

*Démonstration.* Contrairement à la preuve de [HM10], on va utiliser le théorème 5.2.1 pour montrer ce résultat.

Soit  $\mathcal{F}$  un ensemble fini de motifs et  $\mathbf{T} = \mathbf{T}(\mathcal{A}, d, \mathcal{F})$ .

- Un motif  $u$  est *localement admissible* si aucun motif de  $\mathcal{F}$  n'apparaît dans  $u$ . On note  $\tilde{N}_n(\mathcal{F}) = \text{Card}\{u \in \mathcal{A}^{[0, n-1]^d} : u \text{ localement admissible}\}$

- Un motif  $u$  est *globalement admissible* s'il existe  $x \in \mathbf{T}(\mathcal{A}, d, \mathcal{F})$  tel que  $u \sqsubset x$ . On note  $N_n(\mathbf{T}) = \text{Card}\{u \in \mathcal{A}^{[0, n-1]^d} : u \text{ globalement admissible}\}$ . On a  $N_n(\mathbf{T}) \leq \tilde{N}_n(\mathcal{F})$ . On rappelle que

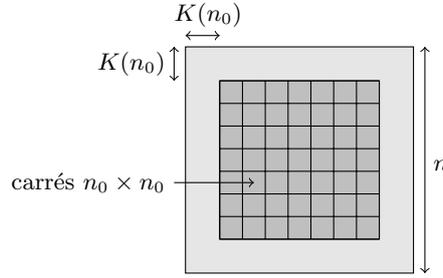
$$h(\mathbf{T}) = \lim_{n \rightarrow \infty} \frac{\log(N_n(\mathcal{F}))}{n^d} = \inf_{n \in \mathbb{N}} \frac{\log(N_n(\mathcal{F}))}{n^d}$$

**Barrière liée à la calculabilité :**

**Lemme 6.1.3.** On a  $\inf_{n \in \mathbb{N}} \frac{\log(N_n(\mathcal{F}))}{n^d} = \inf_{n \in \mathbb{N}} \frac{\log(\tilde{N}_n(\mathcal{F}))}{n^d}$ .

*Démonstration.* On fixe  $n_0 \in \mathbb{N}$ . Pour tout  $u \in \mathcal{A}^{[0, n_0-1]^d}$  localement admissible mais non globalement admissible, il existe  $K(u)$  tel que  $u$  ne peut pas être étendu en un motif de  $[-K(u), K(u)]^d$ . On pose :

$$K(n_0) = \max\{K(u) : u \in \mathcal{A}^{[0, n_0-1]^d} \text{ est localement admissible mais pas globalement}\}$$



On divise  $[0, n-1]^2$  en carré de taille  $n_0 \times n_0$  en laissant une marge de  $K(n_0)$  sur le bord. La bande située sur le bord contient  $n^2 - (n - 2K(n_0))^2$  tuiles et il y a  $\frac{(n-2K(n_0))^2}{n_0^2}$  carrés dans le quadrillage. Comme chaque carré du quadrillage est contenu dans un motif de coté  $2K(n_0) + 1$ , on en déduit qu'ils sont globalement admissibles.

On en déduit que  $\tilde{N}_n(\mathcal{F}) \leq |\mathcal{A}|^{n^2 - (n-2K(n_0))^2} \times N_{n_0}(\mathbf{T})^{\frac{(n-2K(n_0))^2}{n_0^2}}$  d'ou :

$$\frac{1}{n^2} \log(\tilde{N}_n(\mathcal{F})) \leq \frac{4K(n_0)n - 4K(n_0)^2}{n^2} \log(|\mathcal{A}|) + \frac{(n - 2K(n_0))^2}{n^2 n_0^2} \log(N_{n_0}(\mathbf{T}))$$

Par passage à la limite lorsque  $n \rightarrow \infty$ , on obtient  $\inf_{n \in \mathbb{N}} \frac{\log(\tilde{N}_n(\mathcal{F}))}{n^2} \leq \frac{\log(N_{n_0}(\mathbf{T}))}{n_0^2}$ . Comme d'autre part  $\tilde{N}_n(\mathcal{F}) \geq N_n(\mathbf{T})$ , on en déduit que  $\inf_{n \in \mathbb{N}} \frac{\log(N_n(\mathcal{F}))}{n^d} = \inf_{n \in \mathbb{N}} \frac{\log(\tilde{N}_n(\mathcal{F}))}{n^d}$ .  $\square$

Comme  $\tilde{N}_n(\mathcal{F})$  est une suite calculable de nombre, on en déduit que  $h(\mathbf{T})$  est un borne inférieure d'une suite calculable de nombre.

**Réalisation d'un nombre comme entropie d'un sous-shift :**

Pour  $\mathcal{A}_0 \subset \mathcal{A}$  et  $u \in \mathcal{A}^{[0, n-1]^2}$ , on note  $\sharp(\mathcal{A}_0 : u) = \text{Card}\{n \in \mathbb{Z}^2 : u_n \in \mathcal{A}_0\}$ . Pour  $\mathbf{T} \subset \mathcal{A}^{\mathbb{Z}^2}$ , on définit la *densité supérieure* de  $\mathcal{A}_0$  dans  $\mathbf{T}$  par :

$$\delta(\mathcal{A}_0 : \mathbf{T}) = \limsup_{n \rightarrow \infty} \delta_n(\mathcal{A}_0 : \mathbf{T}) \text{ où } \delta_n(\mathcal{A}_0 : \mathbf{T}) = \frac{\max\{\sharp(\mathcal{A}_0 : x_{[0, n-1]^2}) : x \in \mathbf{T}\}}{n^2}$$

**Lemme 6.1.4.** Supposons qu'il existe  $\mathbf{T} = \mathbf{T}(\mathcal{A}, 2, \mathcal{F})$  SFT tel que  $h(\mathbf{T}) = 0$ , alors pour  $\mathcal{A}_0 \subset \mathcal{A}$  il existe un SFT  $\mathbf{T}' = \mathbf{T}'(\mathcal{A}', 2, \mathcal{F}')$  tel que  $h(\mathbf{T}') = \log(2)\delta(\mathcal{A}_0 : \mathbf{T})$ .

*Démonstration.* On pose  $\mathcal{A}' = \mathcal{A} \times \{0, 1\}$  et  $\mathcal{F}' = \mathcal{F} \cup \{(a, b) : a \in \mathcal{A} \setminus \mathcal{A}_0 \text{ et } b = 1\}$ . Ainsi une projection suivant la première coordonnée de  $\mathbf{T}' = \mathbf{T}'(\mathcal{A}', 2, \mathcal{F}')$  donne  $\mathbf{T}$  et si la première coordonnée est dans  $\mathcal{A} \setminus \mathcal{A}_0$ , la deuxième coordonnée vaut nécessairement 0, sinon on a le choix entre 0 ou 1.

Si un motif  $u \in \mathcal{A}^{[0, n-1]^2}$  est globalement admissible pour  $\mathbf{T}$  alors le nombre de motif  $\{u\} \times \{0, 1\}^{[0, n-1]^2}$  globalement admissible pour  $\mathbf{T}'$  est  $2^{n^2 \times \delta_n(\mathcal{A}_0 : u)}$ . On a donc

$$2^{n^2 \times \delta_n(\mathcal{A}_0 : \mathbf{T})} \leq N_n(\mathbf{T}') \leq N_n(\mathbf{T}) 2^{n^2 \times \delta_n(\mathcal{A}_0 : \mathbf{T})}$$

lorsque  $n \rightarrow \infty$  on obtient :  $\log(2)\delta(\mathcal{A}_0 : \mathbf{T}) \leq h(\mathbf{T}') \leq h(\mathbf{T}) + \log(2)\delta(\mathcal{A}_0 : \mathbf{T})$

□

Soit  $\alpha_n = \frac{p_n}{q_n} \in [0, 1]$  une suite décroissante de rationnel énuméré par une machine de Turing. On considère l'ensemble récursivement énumérable suivant :

$$\mathcal{F} = \bigcup_n \{u \in \{0, 1\}^{q_n} : \#(1 : u) > p_n\}$$

Ainsi il existe un SFT  $\mathbf{T}_{\text{Final}} \subset \mathcal{B}^{\mathbb{Z}^2}$  et un facteur lettre à lettre  $\pi : \mathcal{B} \rightarrow \{0, 1\}$  tel que  $\mathbf{T}(\{0, 1\}, 1, \mathcal{F})$  est la sous-action de  $\pi(\mathbf{T}_{\text{Final}})$ . De plus  $\delta(\pi^{-1}(\{1\}) : \mathbf{T}_{\text{Final}}) = \inf_{n \in \mathbb{N}} \alpha_n$ .

Il existe donc un SFT  $\mathbf{T}'_{\text{Final}}$  tel que  $h(\mathbf{T}'_{\text{Final}}) = \log(2)\alpha$ . □

## 6.2 Un S-adic effectif multidimensionnel est sofique

La classe des sous-shifts S-adiques ne peut pas être incluse dans la classe des sous-shifts soifiques par une simple raison de cardinalité : il existe un nombre dénombrables de sous-shifts soifiques, tandis que choisir une suite infinie de substitutions parmi un ensemble fini  $\mathcal{S}$  peut se faire de manière indénombrable. Nous allons montrer dans cette partie que les sous-shifts S-adiques multidimensionnels qui sont soifiques sont exactement ceux dont la suite  $\mathbf{S}$  qui les définit est effective.

Soit  $\mathbf{S} \in \mathcal{S}^{\mathbb{N}}$  une suite de substitutions choisies parmi un ensemble fini  $\mathcal{S}$ . L'inclusion  $\mathbf{T}_{\mathbf{S}} \subset \mathbf{T}_{\mathcal{S}}$  est toujours vérifiée. Par le théorème de Mozes [Moz89], on sait que le sous-shift  $\mathbf{T}_{\mathcal{S}}$  est sofique si l'ensemble  $\mathcal{S}$  vérifie la propriété A, mais il n'y a aucun raison évidente pour que  $\mathbf{T}_{\mathbf{S}}$  le soit aussi.

**Théorème 6.2.1.** *Soit  $\mathcal{S}$  un ensemble fini de substitutions multidimensionnelles non dégénérées et  $\mathbf{S} \in \mathcal{S}^{\mathbb{N}}$  une suite effective de substitutions. Alors le sous-shift  $\mathbf{T}'_{\mathbf{S}}$  est sofique. Si de plus l'ensemble  $\mathcal{S}$  possède la propriété A, alors le sous-shift  $\mathbf{T}_{\mathbf{S}}$  est lui aussi sofique.*

*Démonstration.* Supposons que  $d = 2$ , la démonstration étant similaire lorsque  $d \geq 3$ . Soit  $\mathcal{S}$  un ensemble fini de  $(\mathcal{A}, 2)$ -substitutions non dégénérées. Nous définissons un nouvel alphabet  $\mathcal{A}' = \mathcal{A} \times \mathcal{S}^2$ . A chaque substitution  $s \in \mathcal{S}$  nous associons une  $(\mathcal{A}', 2)$ -substitution  $\tilde{s}$  de même support, définie par

$$\tilde{s} : (a, s_V, s_H) \mapsto \begin{array}{|c|ccc|c|} \hline (s(a)_{(0, \mathbf{k}_2^s(a))}, s, s_H) & s(a)_{(1, \mathbf{k}_2^s(a))}, s, s_H & \dots & s(a)_{(\mathbf{k}_1^s(a)-1, \mathbf{k}_2^s(a))}, s, s_H & (s(a)_{(\mathbf{k}_1^s(a), \mathbf{k}_2^s(a))}, s_V, s_H) \\ \hline (s(a)_{(0, \mathbf{k}_2^s(a)-1)}, s, s) & & & & (s(a)_{(\mathbf{k}_1^s(a), \mathbf{k}_2^s(a)-1)}, s_V, s) \\ \hline \vdots & & (s(a)_{(i, j)}, s, s) & & \vdots \\ \hline (s(a)_{(0, 1)}, s, s) & & & & (s(a)_{(\mathbf{k}_1^s(a), 1)}, s_V, s) \\ \hline (s(a)_{(0, 0)}, s, s) & (s(a)_{(1, 0)}, s, s) & \dots & (s(a)_{(\mathbf{k}_1^s(a)-1, 0)}, s, s) & (s(a)_{(\mathbf{k}_1^s(a), 0)}, s_V, s) \\ \hline \end{array}$$

Toutes ces substitutions  $\tilde{s}$  forment un ensemble fini  $\tilde{\mathcal{S}} = \{\tilde{s} : s \in \mathcal{S}\}$ . Soit  $\mathbf{S} = (s_i)_{i \in \mathbb{N}} \in \mathcal{S}^{\mathbb{N}}$  une suite effective, nous pouvons donc lui associer la suite effective  $\tilde{\mathbf{S}} = (\tilde{s}_i)_{i \in \mathbb{N}} \in \tilde{\mathcal{S}}^{\mathbb{N}}$ . L'objectif

de ces substitutions  $\tilde{s}$  est de garder en mémoire la suite des substitutions précédemment appliquées. Pour cela une substitution  $s$  appliquée à un certain niveau d'itération est gardée en mémoire et transférée aux niveaux suivants grâce aux substitutions  $s_V$  (pour transfert vertical) et  $s_H$  (pour transfert horizontal) de l'alphabet  $\mathcal{A}'$ . Ces transferts sont visibles sur les dernières ligne et colonne d'un motif  $\tilde{s}(a, s_V, s_H)$ .

Considérons la fonction 1-bloc  $\pi : \mathcal{A}' \rightarrow \mathcal{A}$  qui ne conserve que la lettre de  $\mathcal{A}$  et  $\pi_V : \mathcal{A}' \rightarrow \mathcal{S}$  (respectivement  $\pi_H : \mathcal{A}' \rightarrow \mathcal{S}$ ) la fonction 1-bloc qui ne conserve que la substitution  $s_V \in \mathcal{S}$  (respectivement  $s_H \in \mathcal{S}$ ) d'un élément  $(a, s_V, s_H) \in \mathcal{A}'$ .

On a  $\mathbf{T}_S = \pi(\mathbf{T}_{\tilde{S}})$ .

De plus, le sous-shift  $\Sigma = \mathbf{SA}_{(1,0,\dots,0)\mathbb{Z}}(\pi_V(\mathbf{T}_{\tilde{S}}))$  est effectif. Par le Théorème 5.2.1, il existe un sous-shift de type fini de dimension  $d$  sur un alphabet  $\mathcal{B}$ , que l'on notera  $\mathbf{T}_\Sigma$ , et une fonction de bloc  $\pi_\Sigma : \mathcal{B}^{\mathbb{Z}^d} \rightarrow \mathcal{S}^{\mathbb{Z}^d}$  tels que  $\Sigma = \pi_\Sigma(\mathbf{SA}_{(1,0,\dots,0)\mathbb{Z}}(\mathbf{T}_\Sigma))$ .

Si nous considérons une configuration du sous-shift  $\mathbf{T}_{\tilde{S}}$ , toute substitution qui y apparaît peut être choisie dans l'ensemble  $\mathcal{S}$ , à condition qu'elle soit compatible avec la configuration. Mais des substitutions différentes peuvent apparaître sur un même niveau. Ceci n'est pas cohérent avec la définition des sous-shifts S-adiques. Pour que cela le devienne, il suffit d'assurer que dans une configuration  $x \in \mathbf{T}_{\tilde{S}}$ , la même substitution apparaît sur chaque ligne de  $\pi_V(x)$  et la même substitution apparaît sur chaque colonne de  $\pi_H(x)$ . Nous définissons le sous-shift  $\tilde{\mathbf{T}}_{\tilde{S}}$  par :

$$\tilde{\mathbf{T}}_{\tilde{S}} = \{x \in \mathbf{T}_{\tilde{S}} : \forall (i, j) \in \mathbb{Z}^2, \pi_H(x)_{(i,j)} = \pi_H(x)_{(i,j+1)} \text{ et } \pi_V(x)_{(i,j)} = \pi_V(x)_{(i+1,j)}\},$$

et les conditions imposées permettent de déduire que :  $\tilde{\mathbf{T}}_{\tilde{S}} = \bigcup_{\tilde{S} \in \tilde{S}^{\mathbb{N}}} \mathbf{T}_{\tilde{S}} \subset \mathbf{T}_{\tilde{S}}$ .

Considérons finalement le sous-shift suivant :

$$\mathbf{T}_{\text{Final}} = \{(x, s) \in \tilde{\mathbf{T}}_{\tilde{S}} \times \mathbf{T}_\Sigma : \forall (i, j) \in \mathbb{Z}^2, \pi_V(x)_{(i,j)} = \pi_\Sigma(s)_{(i,j)}\}.$$

Par le Corollaire 3.2.3, nous savons que le sous-shift  $\mathbf{T}_{\tilde{S}}$  est sofique, il est en donc de même pour  $\tilde{\mathbf{T}}_{\tilde{S}}$ . Par construction,  $\mathbf{T}_{\text{Final}}$  est aussi un sous-shift sofique. Considérons la fonction 1-bloc  $\pi_{\text{Final}} : \mathbf{T}_{\text{Final}} \rightarrow \mathcal{A}'^{\mathbb{Z}^d}$  qui ne conserve que les lettres de l'alphabet  $\mathcal{A}'$ .

**Proposition 6.2.2.**  $\pi_{\text{Final}}(\mathbf{T}_{\text{Final}}) = \mathbf{T}_{\tilde{S}}$

*Démonstration.* Étant donnée une configuration  $x \in \mathbf{T}_{\tilde{S}}$ , il est facile de construire l'élément correspondant dans  $\mathbf{T}_{\text{Final}}$ . Réciproquement, soit  $x_{\text{Final}} \in \mathbf{T}_{\text{Final}}$ . Quitte à remplacer les substitutions de l'ensemble  $\mathcal{S}$  par la composition de deux substitutions de  $\mathcal{S}$ , nous pouvons supposer que pour toute substitution  $s \in \mathcal{S}$  et pour toute lettre  $a \in \mathcal{A}$ ,  $\mathbf{k}_1^s(a), \mathbf{k}_2^s(a) \geq 2$  (car les substitutions ne sont pas dégénérées).

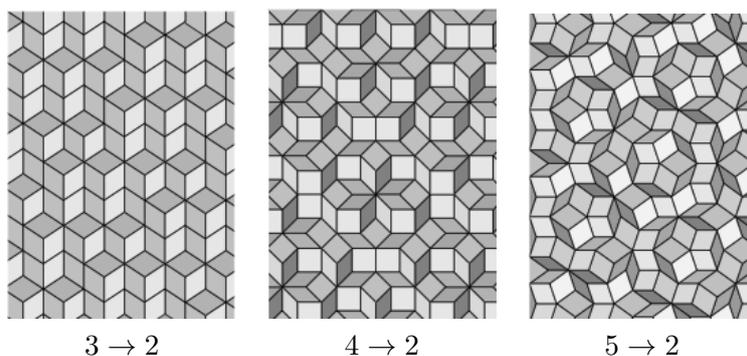
La partie correspondant à  $\tilde{\mathbf{T}}_{\tilde{S}}$  dans la configuration  $x_{\text{Final}}$  nous assure que  $\pi_{\text{Final}}(x_{\text{Final}})$  est un élément de l'un des sous-shifts  $\mathbf{T}_{\tilde{S}}$  pour une certaine suite  $\tilde{S}' \in \tilde{S}^{\mathbb{N}}$ . De plus, la condition liant les parties correspondant à  $\tilde{\mathbf{T}}_{\tilde{S}}$  avec celle correspondant à  $\mathbf{T}_\Sigma$  garantit que  $\mathbf{S}' = \mathbf{S}$ . Ainsi la substitution  $s_0$  est la seule qui soit systématiquement répétée au moins deux fois, car nous avons supposé que  $\mathbf{k}_1^s, \mathbf{k}_2^s \geq 2$ . Le même raisonnement appliqué à une pré-image de  $x_{\text{Final}}$  par  $s_0$  nous permet de retrouver la substitution  $s_1$ , et ainsi de suite pour les autres éléments de la suite  $\mathbf{S}$ .  $\square$

Nous avons donc montré que  $\mathbf{T}_S$  est sofique. La réciproque à ce théorème est vrai si toutes les substitutions de  $\mathcal{S}$  sont à dérivation unique.  $\square$

## 6.3 Pavages obtenus par coupé-projection réalisables par des sofiques

### 6.3.1 Position du problème

Soient  $\mathbf{v}_1, \dots, \mathbf{v}_n$  des vecteurs de  $\mathbb{R}^d$  non colinéaire deux à deux avec  $n > d > 0$ .



On définit les notions suivantes :

- Une *tuile*  $n \rightarrow d$  est un paralléloétope engendré par  $d$  des  $\mathbf{v}_i$  vecteurs, il y a donc  $\binom{n}{d}$  tuiles.
- Un *pavage*  $n \rightarrow d$  est un pavage de  $\mathbb{R}^d$  par des tuiles  $n \rightarrow d$ .
- L'ensemble  $\mathbf{X}_{n \rightarrow d}$  de tous les pavages de  $\mathbb{R}^d$  par des tuiles  $n \rightarrow d$ , c'est l'espace complet *espace complet des pavages*  $n \rightarrow d$ .
- Étant donné un pavage  $n \rightarrow d$ , on associe à un de ces sommets un vecteur de  $\mathbb{Z}^n$ . Puis on modifie la  $k^{\text{ème}}$  coordonnée lorsqu'on se déplace suivant le  $k^{\text{ème}}$  vecteur. Les sommets de la tuile  $n \rightarrow d$  sont donc envoyés sur les sommets de  $[0, 1]^n$ . Ainsi, le pavage  $n \rightarrow d$  est envoyé sur une surface discrète  $\mathbb{R}^n$  : c'est la *surface relevée* du pavage (cf figure 6.1).

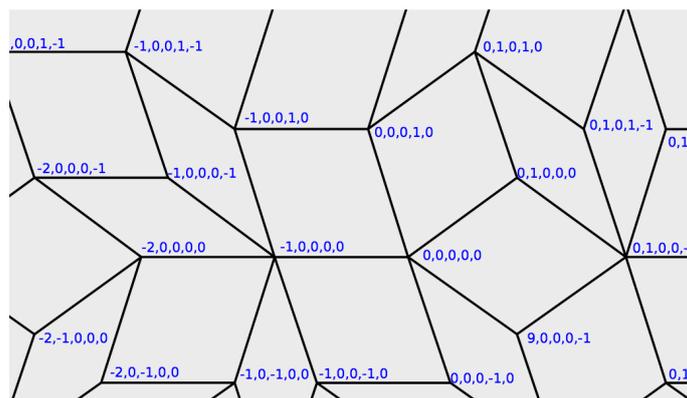
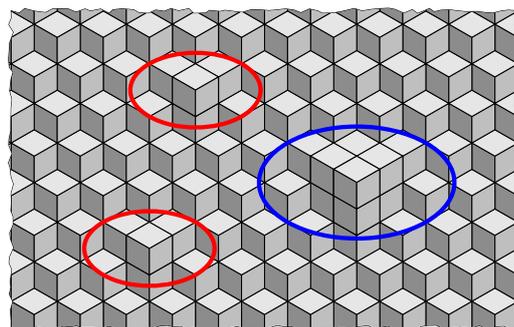
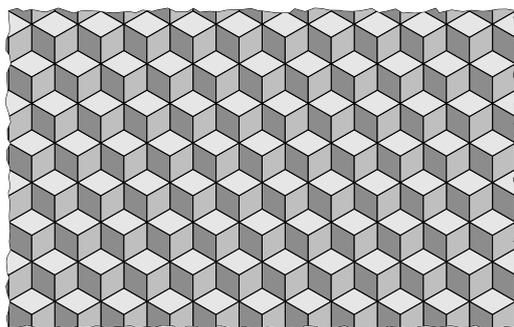
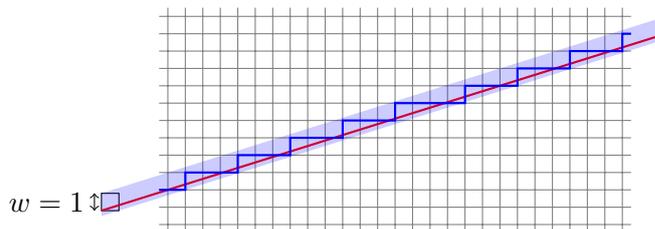


FIGURE 6.1 – Coordonnées de la surface relevée associé à un pavage.

Un espace de pavage  $n \rightarrow d$   $\mathbf{T} \subset \mathbf{X}_{n \rightarrow d}$  est un *espace planaire de pavages* s'il existe un espace vectoriel  $d$ -dimensionnel  $V \subset \mathbb{R}^n$ , la *penne* et un entier positif  $w$ , l'*épaisseur*, tel que le pavage  $t \in \mathbf{T}$  peut être relevé dans l'ensemble  $V + [0, w]^n$ .

Le cas  $w = 1$  correspond au *pavage planaire parfait*.

Le cas  $w \geq 2$  correspond au *pavage planaire faible*.



Un  $n \rightarrow d$ -motif de taille  $r$  d'un pavage  $t \in \mathbf{X}_{n \rightarrow d}$  est l'ensemble des tuiles contenues dans une boule de rayon  $r > 0$ . Pour  $\mathbf{T} \subset \mathbf{X}_{n \rightarrow d}$ , on note  $\mathcal{P}_r(\mathbf{T})$  l'ensemble des  $n \rightarrow d$ -motifs de taille  $r$  de chaque pavage de  $\mathbf{T}$ . L'ensemble des pavage dont les  $n \rightarrow d$ -motifs  $\mathcal{F}$  sont interdits est :

$$\mathbf{T}_{\mathcal{F}} = \{t \in \mathbf{X}_{n \rightarrow d} : \text{aucun motif de } \mathcal{F} \text{ apparait dans } t\}$$

On dit que  $\mathbf{T}$  est un *espace de pavage de type fini* si il existe  $\mathcal{F}$  fini tel que  $\mathbf{T} = \mathbf{T}_{\mathcal{F}}$  on dit aussi que  $\mathbf{T}$  admet des *règles locales*.

On dit qu'un espace de pavage  $\mathbf{T}$  admet des *règles locales colorées* s'il correspond a un espace de pavage obtenu en coloriant les arêtes des tuiles  $n \rightarrow d$  et en autorisant à placer deux tuiles côte à côte si et seulement si les couleurs sur les deux arêtes correspondent. Des règles locales décrivant un espace de pavage planaire faible sont dits *naturelles* si l'espace de pavage planaire parfait vérifie ces règles locales.

On peut se demander pour quel espace vectoriel  $V$  de dimension  $d$  on peut définir un pavage planaire  $n \rightarrow d$  (parfait ou faible) de pente  $V$  à l'aide de règles locales ou règles locales colorées dans ce cas on dit que  $V$  admet des *règles locales colorées ou pas*.

### 6.3.2 Résultats principaux

On considère le plan  $\mathbb{R}(u_1, \dots, u_n) + \mathbb{R}(v_1, \dots, v_n)$  où  $u_k = \cos\left(\frac{2k\pi}{n}\right)$  and  $v_k = \sin\left(\frac{2k\pi}{n}\right)$ , le pavage  $n \rightarrow 2$  obtenu s'appelle un *pavage à n-symétries*.

Dans la littérature, on trouve de nombreuses références à ce problème :

Pavage	règles locales sans décoration	règles locales avec décoration
5, 10-symétries	forte	forte [Pen78]
8-symétries	pas de règles naturelles [Bur88]	forte [Tha95]
12-symétries	pas de règles naturelles [TPS92]	forte [Soc89]
$n$ -symétries (avec 4 qui ne divise pas $n$ )	faible [Soc90]	forte ?
pente quadratique dans $\mathbb{R}^4$	faible [Lev88]	forte [TPS92]
non algébrique	pas de règles naturelles [Le97]	?

Comme le montre le tableau précédent, jusqu'à présent on s'intéresse à des conditions algébriques. En fait le point de vu donné par le théorème 5.2.1 permet de construire des règles locales pour des plans calculables.

Un vecteur  $\mathbf{v} \in \mathbb{R}^n$  est *calculable* si il existe une fonction calculable  $f : \mathbb{N} \rightarrow \mathbb{Q}^n$  tel que  $\|\mathbf{v} - f(n)\|_\infty \leq \frac{1}{2^n}$  pour tout  $n \in \mathbb{N}$ . Un espace vectoriel  $V \subset \mathbb{R}^n$  de dimension  $d$  est *calculable* si il existe un ensemble de  $d$  vecteurs calculables qui engendre  $V$ .

**Théorème 6.3.1.** [[FS12]] *Un espace vectoriel  $d$ -dimensionnel  $V$  admet des  $n \rightarrow d$ -règles locales colorées faibles (d'épaisseur 3) pour  $n > d$  si et seulement si il est calculable.*

**Théorème 6.3.2.** [[FS12]] *Un espace vectoriel  $d$ -dimensionnel  $V$  admet des  $n \rightarrow d$ -règles locales faibles (d'épaisseur 4) pour  $n > d$  si et seulement si il est calculable.*

Pour ce résultat, bien sûr les règles locales obtenues ne sont pas naturelles.

### 6.3.3 Idée de preuve

#### Barrière liée à la calculabilité :

Soit  $\mathbf{S} = \{\vec{x} \in \mathbb{R}^n : \|\mathbf{x}\|_\infty = 1\}$ , étant donné  $V$  et  $W$  deux espaces vectoriels de dimension  $d$  de  $\mathbb{R}^n$ , on définit la distance :

$$\tilde{d}(V, W) = \max \left\{ \sup_{\vec{v} \in \mathbf{S} \cap V} \inf_{\vec{w} \in W} \|\vec{v} - \vec{w}\|_\infty ; \sup_{\vec{w} \in \mathbf{S} \cap W} \inf_{\vec{v} \in V} \|\vec{v} - \vec{w}\|_\infty \right\}.$$

De manière équivalente, l'espace vectoriel  $V$  est *calculable* s'il existe une machine de Turing qui sur l'entrée  $n \in \mathbb{N}$  donne une base rationnelle qui définit  $W_n$  un espace vectoriel de dimension  $d$  tel que  $\tilde{d}(V, W_n) \leq \frac{1}{n}$ .

On cherche à écrire un algorithme pour obtenir la pente connaissant les règles locales de l'espace de pavages planaire  $\mathbf{T}$ , l'épaisseur  $w$  et un entier  $n$ .

#### Algorithme :

- $r := 2wn$  et  $d := 1$
- Tant que  $d \geq \frac{1}{2n}$  faire :
  - on énumère  $\mathcal{P}_r(\mathbf{T})$ , l'ensemble des motifs autorisés par ces règles locales de diamètre  $r$  centrés sur 0 (ceci prend un temps fini mais exponentiel en  $r$ )
  - on énumère  $\mathbb{X}_r$ , l'ensemble des espaces vectoriels de dimension  $d$  associés aux sommets du bord lorsqu'on relève  $P_r$  et on calcule

$$d = \max_{W_1, W_2 \in \mathbb{X}_r} \tilde{d}(W_1, W_2)$$

- Sortie : un élément de  $W \in \mathbb{X}_r$

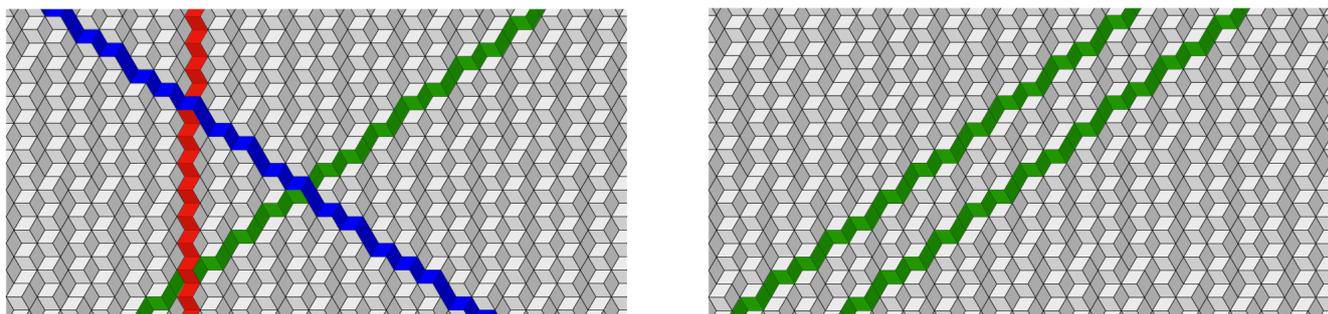
Cet algorithme s'arrête car pour un  $r$  suffisamment grand, tout les espaces vectoriels de  $\mathbb{X}_r$  sont proche de  $V$  car sinon par compacité on aurait plusieurs pentes pour le pavage planaire  $\mathbf{T}$ .

L'algorithme aboutit à une approximation de la pente de  $\mathbf{T}$ . En effet, il existe  $W' \in \mathbb{X}_r$  tel que  $\tilde{d}(W', V) \leq \frac{w}{r}$ , ainsi

$$\tilde{d}(W, V) \leq \tilde{d}(W, W') + \tilde{d}(W', V) \leq \frac{1}{2n} + \frac{w}{r} \leq \frac{1}{n}$$

### Réalisation d'un espace vectoriel calculable :

On va démontrer le résultat pour un pavage  $3 \rightarrow 2$ . Pour un pavage  $3 \rightarrow 2$  planaire parfait, il y a des bandes codant des mots sturmiens qui se croisent et deux lignes parallèles codent deux mot sturmiens de même pente.



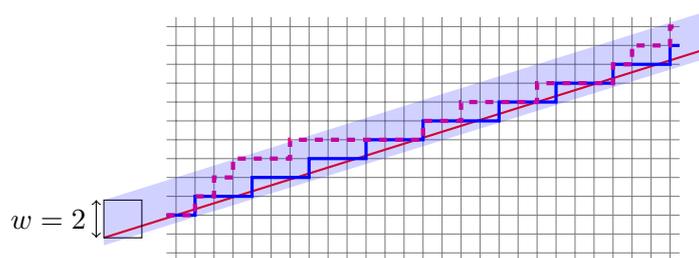
On définit le *mot sturmien*  $s_{\rho, \alpha} \in \{0, 1\}^{\mathbb{Z}}$  de pente  $\alpha \in [0, 1]$  et d'origine  $\rho$  par

$$s_{\rho, \alpha}(n) = 0 \iff (\rho + n\alpha) \bmod 1 \in [0, 1 - \alpha).$$

Pour  $x, y \in \{0, 1\}^{\mathbb{Z}}$  on définit  $d(x, y) := \sup_{p \leq q} ||x_p x_{p+1} \dots x_q|_0 - |y_p y_{p+1} \dots y_q|_0|$ . Deux mots sturmiens de même pente sont à distance au plus égal à 1.

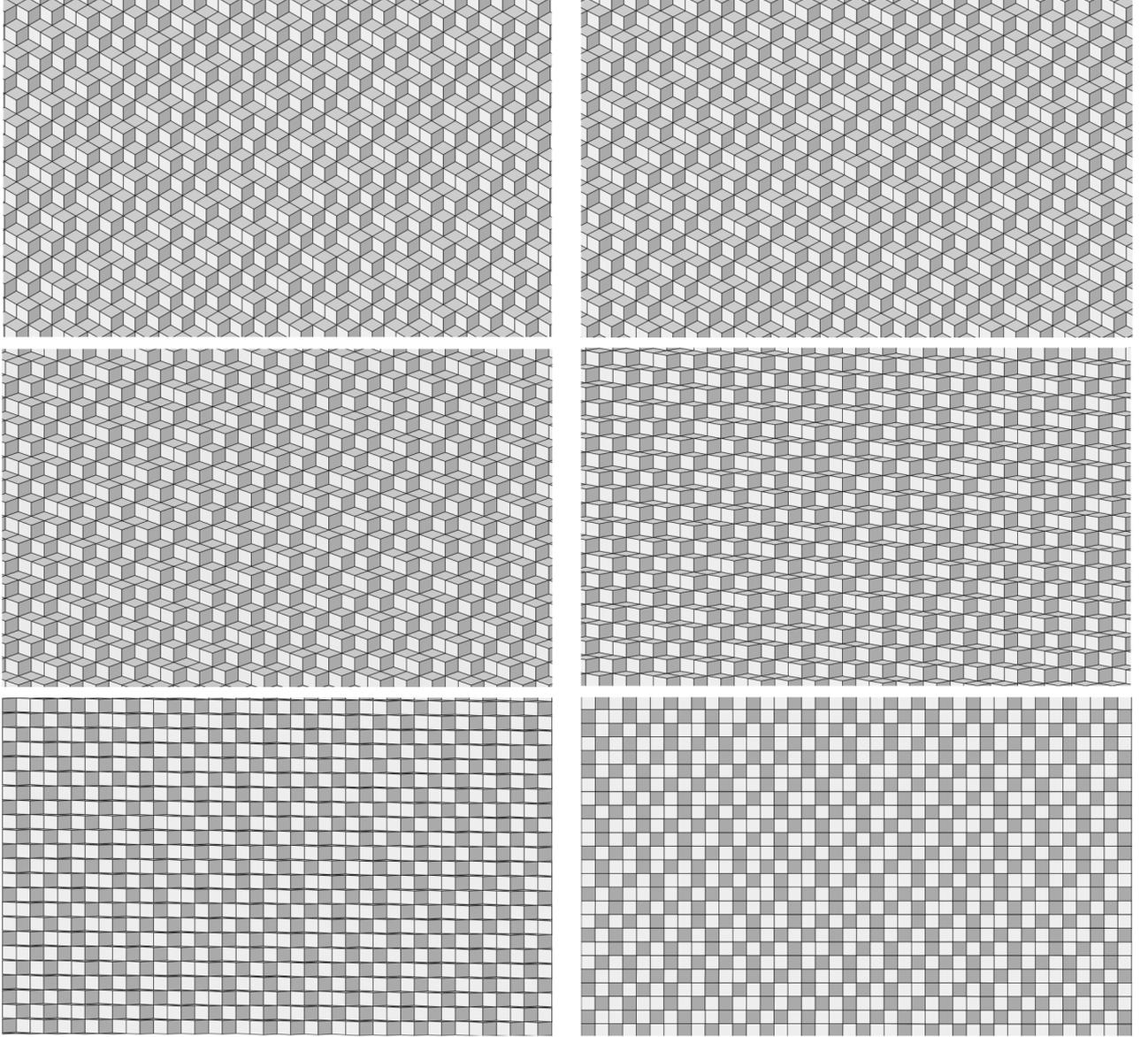
On dit que  $x \in \{0, 1\}^{\mathbb{Z}}$  est un *mot quasi-sturmien de pente  $\alpha$*  si  $d(x, s_{\rho, \alpha}) \leq 1$ .

On remarque que  $x, y \in \{0, 1\}^{\mathbb{Z}}$  vérifient  $d(x, y) \leq 1$  si et seulement si on peut obtenir  $x$  en modifiant  $y$  par les remplacements de lettres  $0 \rightarrow 1$  ou  $1 \rightarrow 0$ , de tel sorte qu'il y ait une alternance des types de remplacement.



$s_{\rho, \alpha}$  : ...001000100010001000100010001000100010...  
 Quasi-Sturmien : ...0010101000100000001001000010001001010...  
 Codage : ...00000111111111110000000110001000100111...

On appelle  $\mathbf{v}_i$ -*bande* d'un pavage  $3 \rightarrow 2$  une suite de tuiles adjacentes suivant l'arrête  $\mathbf{v}_i$ . Si on ne conserve que ces bandes on obtient un sous-shift  $\mathbf{T}_{\mathbf{v}_i} \subset \{0, 1\}^{\mathbb{Z}^2}$ , cela revient à regarder la surface discrète de telle sorte que l'on ne voit que deux faces des cubes  $[0, 1]^3$  qui forment la surface :



On note  $S_\alpha = \{x \in \{0, 1\}^{\mathbb{Z}^2} : \forall i \in \mathbb{Z}, \exists \rho, x_{\mathbb{Z} \times \{i\}} = s_{\rho, \alpha}\}$  le *sous-shift sturmien* et  $Q_\alpha = \{x \in \{0, 1\}^{\mathbb{Z}^2} : \forall i \in \mathbb{Z}, d(x_{\mathbb{Z} \times \{i\}}, s_{\rho, \alpha}) \leq 1\}$  le *sous-shift quasi-sturmien*. Il existe  $\alpha$  tel que

$$\mathbf{T}_{v_i} \subset S_\alpha \subset Q_\alpha.$$

**Lemme 6.3.3.** *Le sous-shift quasi-sturmien de pente  $\alpha$   $Q_\alpha$  est sofique.*

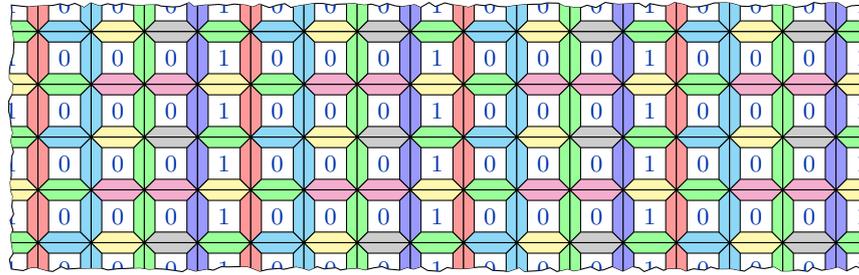
Si  $\alpha$  est calculable alors  $\{s_{\alpha, \rho} \in \{0, 1\}^{\mathbb{Z}} : \rho \in \mathbb{R}\}$  est un sous-shift effectif. Ainsi il existe un SFT  $\mathbf{T}_{\{0,1\} \times \mathcal{B}, \mathcal{F}} \subset (\{0, 1\} \times \mathcal{B})^{\mathbb{Z}^2}$  tel que  $\pi_1(\mathbf{T}_{\{0,1\} \times \mathcal{B}, \mathcal{F}}) = \{x \in \{0, 1\}^{\mathbb{Z}^2}, \exists \rho \in \mathbb{R}, \forall m \in \mathbb{Z}, w_{\cdot, m} = s_{\alpha, \rho}\}$ .

On considère SFT  $\tilde{Q}_\alpha \subset (\{0, 1\} \times \mathcal{B} \times \{0, 1\})^{\mathbb{Z}^2}$  tel que :

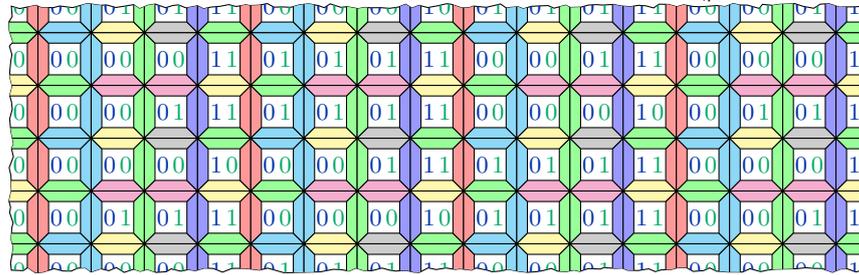
$$x \in \tilde{Q}_\alpha \iff \begin{cases} \pi_{12}(x) \in \mathbf{T}_{\mathcal{B}, \mathcal{F}}, \\ \pi_3(x_{m,n}) = 0 \text{ and } \pi_3(x_{m,n+1}) = 1 \Rightarrow \pi_1(x_{m,n}) = 0, \\ \pi_3(x_{m,n}) = 1 \text{ and } \pi_3(x_{m,n+1}) = 0 \Rightarrow \pi_1(x_{m,n}) = 1. \end{cases}$$

On définit  $\pi(x)_{m,n} = \begin{cases} \pi_1(x_{m,n}) & \text{si } \pi_3(x_{m,n}) = \pi_3(x_{m,n+1}), \\ 1 - \pi_1(x_{m,n}) & \text{dans les autres cas.} \end{cases}$ , ainsi

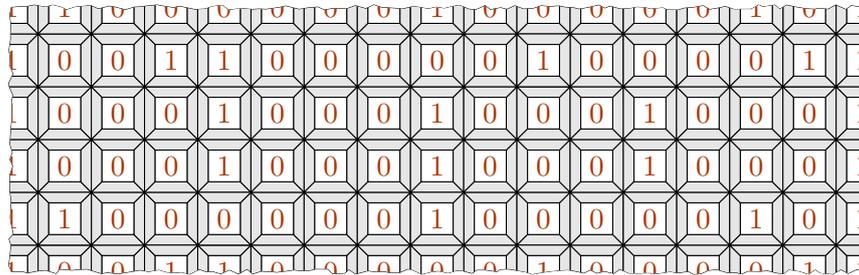
$$\pi(\tilde{Q}_\alpha) = Q_\alpha = \{x \in \{0,1\}^{\mathbb{Z}^2}, \forall m \in \mathbb{Z}, d(x_{(\cdot,m)}, s_{\alpha,0}) \leq 1\}.$$



Chaque ligne est le même mot sturmien  $s_{\alpha,p}$



Sur chaque ligne on rajoute un codage valide de manière indépendante.



Après projection du facteur  $\pi$ , chaque ligne est un mot quasi-sturmien de pente  $\alpha$  indépendant des autres.

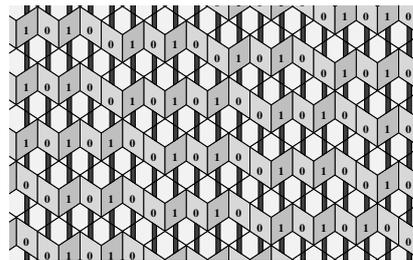
Chaque tuile de  $\tilde{Q}_\alpha$  peut être vue comme un tuile de Wang. On construit l'ensemble  $\tau_\alpha^{\mathbf{v}_3}$  de tuiles  $3 \rightarrow 2$  coloriées de la manière suivante :



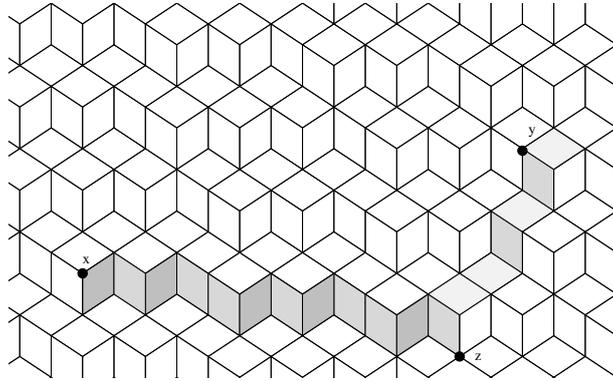
```

0 1 0 1 0
1 0 1 0 0 1 0 1 0 0 1 0 0 1 0 0 1 0 1 0
1 0 1 0 0 1 0 1 0 1 0 0 1 0 0 1 0 1 0 0 1 0
1 0 1 0 1 0 0 0 1 0 1 0 0 0 1 0 1 0 0 1 0
0 0 1 0 1 0 0 0 1 0 1 0 0 0 1 0 1 0 1 0 0 0
0 0 1 0 1 0

```



Ainsi,  $\tau_\alpha^{\mathbf{v}_3}$  force exactement des  $\mathbf{v}_3$ -bandes à être de pente  $\alpha$ . Comme précédemment, on construit les jeux de tuiles  $\tau_\beta^{\mathbf{v}_2}$  et  $\tau_{\alpha/\beta}^{\mathbf{v}_1}$  et on considère le pavage  $3 \rightarrow 2$  formé avec  $\tau_{1,\alpha,\beta} = \tau_{\alpha/\beta}^{\mathbf{v}_1} \times \tau_\beta^{\mathbf{v}_2} \times \tau_\alpha^{\mathbf{v}_3}$ .



Deux sommets de ce pavage sont reliés par un chemin formé par deux de ces bandes : on a donc un pavage planaire de pente orthogonale à  $(1, \alpha, \beta)$  et d'épaisseur au plus 3. On en déduit le théorème 6.3.1

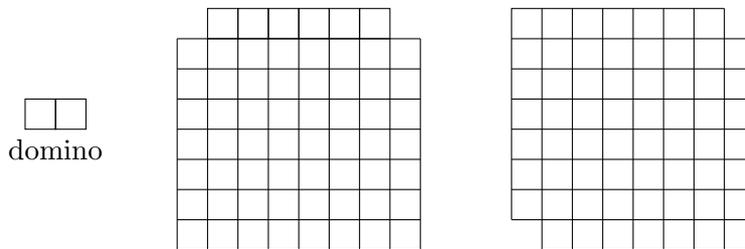
Étant donné une pente, il est possible de substituer chaque tuile par une "meta" tuile arbitrairement large. Ainsi les décorations peuvent être codées par des fluctuations et l'épaisseur n'augmente pas plus de 1. Cette idée permet de démontrer le théorème 6.3.2.

# Chapitre 7

## Exercices

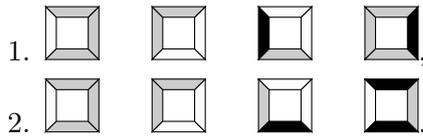
### 7.1 Dynamique symbolique

**Exercice 1.** Est-il possible de paver les échiquiers partiels suivants avec des dominos  $1 \times 2$  ?



**Exercice 2.** Prouver qu'il n'est pas possible de paver une rectangle  $10 \times 14$  en utilisant 35 copies de rectangles  $1 \times 4$ .

**Exercice 3.** Déterminer si les jeux de tuiles de Wang suivants pavent le plan :



**Exercice 4.** Un langage  $\mathcal{L} \subset \bigcup_{n \in \mathbb{N}} \mathcal{A}^{[-n, n]^d}$  est *extensible* si

- pour tout  $u \in \mathcal{L}$  et pour tout  $v \sqsubset u$  de support  $[-m, m]^d$  avec  $m \in \mathbb{N}$  on a  $v \in \mathcal{L}$  ;
- pour tout  $u \in \mathcal{L} \cap \mathcal{A}^{[-n, n]^d}$  il existe  $v \in \mathcal{L} \cap \mathcal{A}^{[-n-1, n+1]^d}$  tel que  $v \in \mathcal{L}$  et  $u \sqsubset v$ .

On définit  $\mathbf{T}(\mathcal{L}) = \{x \in \mathcal{A}^{\mathbb{Z}^d} : \forall n \in \mathbb{N}, x_{[-n, n]^d} \in \mathcal{L}\}$ . Montrer les propriétés suivantes :

1. Si  $\mathcal{L} \subset \bigcup_{n \in \mathbb{N}} \mathcal{A}^{[-n, n]^d}$  est un langage extensible alors  $\mathbf{T}(\mathcal{L})$  est un sous-shift.
2. Si  $\mathcal{L} \subset \bigcup_{n \in \mathbb{N}} \mathcal{A}^{[-n, n]^d}$  est un langage *extensible* alors  $\mathcal{L}(\mathbf{T}(\mathcal{L})) = \mathcal{L}$ .
3. Si  $\mathbf{T} \subset \mathcal{A}^{\mathbb{Z}^d}$  est un sous-shift alors  $\mathbf{T}(\mathcal{L}(\mathbf{T})) = \mathbf{T}$ .
4. Si  $x \in \mathcal{A}^{\mathbb{Z}^d}$  alors  $\mathcal{L}(x)$  est un langage extensible et

$$\mathbf{T}(\mathcal{L}(x)) = \overline{\mathcal{O}(x)} = \{y \in \mathcal{A}^{\mathbb{Z}^d} : \forall n \geq 0, \exists i \in \mathbb{Z}^d, y_{U_n} = x_{i+U_n}\}.$$

**Exercice 5.** Soit  $F : \mathcal{A}^{\mathbb{Z}^d} \rightarrow \mathcal{A}^{\mathbb{Z}^d}$ . On définit  $\Lambda_F = \bigcap_{n \in \mathbb{N}} F^n(\mathcal{A}^{\mathbb{Z}^d})$ .

1. Montrer que  $\Lambda_F$  est non vide.

2. Montrer que  $\Lambda_F$  est fini si et seulement s'il existe  $n \in \mathbb{N}$  et  $c \in \mathcal{A}^{\mathbb{Z}^d}$  tel que  $F^n(\mathcal{A}^{\mathbb{Z}^d}) = \{c\}$ .
3. Soit  $C \subset \mathcal{A}^{\mathbb{Z}^d}$  un fermé, en appliquant le théorème de Baire à la suite de fermés  $(F^{-n}(C))_{n \in \mathbb{N}}$ , montrer que soit il existe  $n \in \mathbb{N}$  tel que  $F^n(\mathcal{A}^{\mathbb{Z}^d}) \subset C$ , soit il existe  $D$  un  $G_\delta$  dense  $\sigma$ -invariant tel que  $C \cap \bigcup_{n \in \mathbb{N}} F^n(D) = \emptyset$ . En appliquant ce résultat à l'ensemble  $\Lambda_F$ , montrer que soit il existe  $n \in \mathbb{N}$  tel que  $\Lambda_F = F^n(\mathcal{A}^{\mathbb{Z}^d})$ , soit il existe une configuration dont l'orbite ne rentre jamais dans  $\Lambda_F$ .

**Exercice 6.** Soit  $\mathbf{T} \subset \mathcal{A}^{\mathbb{Z}^d}$  un sous-shift de type fini, montrer que il existe  $x \in \mathbf{T}$  tel que toute lettre apparaissant dans  $x$  apparaisse une infinité de fois.

**Exercice 7.** Considérons un jeu de tuile de Wang tel qu'on paver une ligne horizontale en respectant les couleurs et tels que pour tout segment fini, si on considère les couleurs sur la partie supérieure de la ligne horizontale, il existe un segment qui a le même motif sur la partie inférieure de la ligne horizontale.

Montrer qu'un tel jeu de tuile de Wang pave le plan.

**Exercice 8.** Soit  $\mathcal{F}$  un ensemble de motifs tel que

$$\mathbf{T}(\mathcal{A}, d, \mathcal{F}) = \{x \in \mathcal{A}^{\mathbb{Z}^d} : p \text{ n'apparaît pas dans } x \text{ pour tout } p \in \mathcal{F}\} = \emptyset.$$

Montrer qu'il existe un ensemble fini de motifs  $\mathcal{F}' \subset \mathcal{F}$  tel que  $\mathbf{T}(\mathcal{A}, d, \mathcal{F}') = \emptyset$ .

**Exercice 9.** Soit  $x \in \mathcal{A}^{\mathbb{Z}^d}$ , montrer que  $\mathcal{O}(x)$  est un sous-shift si et seulement si  $x$  est périodique.

**Exercice 10.** 1. Montrer que l'ensemble des configurations périodiques dans toutes directions de  $\mathcal{A}^{\mathbb{Z}^d}$ , c'est à dire

$$P = \left\{ x \in \mathcal{A}^{\mathbb{Z}^d} : \exists (p_1, \dots, p_d) \in \mathbb{N}^d \text{ tel que } \sigma^{p_1 n_1 \mathbf{e}_1 + \dots + p_d n_d \mathbf{e}_d}(x) = x \forall (n_1, \dots, n_d) \in \mathbb{Z}^d \right\},$$

est dense dans  $\mathcal{A}^{\mathbb{Z}^d}$ .

2. Soit  $F : \mathcal{A}^{\mathbb{Z}^d} \rightarrow \mathcal{B}^{\mathbb{Z}^d}$  une fonction continue qui commute avec le shift. Montrer que si  $F$  est injective alors elle est surjective sur  $P$ .
3. Montrer que si  $F$  est injective alors elle est surjective.

**Exercice 11.** On considère  $\mathcal{P}$  un ensemble fini de tuiles de Wang où les couleurs sont données par des flèches entrantes et sortantes. Il peut y avoir plusieurs flèches par côté et deux tuiles peuvent se coller suivant une arête s'il y a le même nombre d'arêtes entrantes de chaque côté et le même nombre d'arêtes sortantes de chaque côté.

Montrer que si dans chaque tuile de  $\mathcal{P}$  il y a plus d'arête sortante que d'arêtes entrantes alors  $\mathcal{P}$  ne peut pas paver le plan.

**Exercice 12.** Soit  $\mathbf{T} \subset \{0, 1\}^{\mathbb{Z}^2}$  l'ensemble des configurations tel que tout motif  $2 \times 2$  contient exactement deux 0 et deux 1.

1. Déterminer si  $\mathbf{T}$  est un sous-shift, un sofique, un sous-shift de type fini.
2. Prouver que toute configuration de  $\mathbf{T}$  est périodique au moins dans une direction.
3. Est ce que  $\mathbf{T}$  est transitif?
4. Donner des exemples de sous-shift minimaux de  $\mathbf{T}$ .

**Exercice 13.** Soit  $\mathbf{T} \subset \{0, 1\}^{\mathbb{Z}^2}$  l'ensemble des configurations définit par

$$\mathbf{T} = \left\{ x \in \{0, 1\}^{\mathbb{Z}^2} \text{ il existe } i, j \in \mathbb{Z}, n \in \mathbb{N} \text{ tel que } x_{i', j'} = 1 \iff i \leq i' \leq i + n \text{ et } j \leq j' \leq j + n \right\}.$$

En d'autres termes, une configuration de  $\mathbf{T}$  est un carré formé de 1 sur un fond de 0.

1. Est ce que  $\mathbf{T}$  est un sous-shift ?
2. Décrire les éléments de la clôture  $\overline{\mathbf{T}}$ .
3. Est ce que  $\overline{\mathbf{T}}$  est un sous-shift, un sofique, un sous-shift de type fini ?

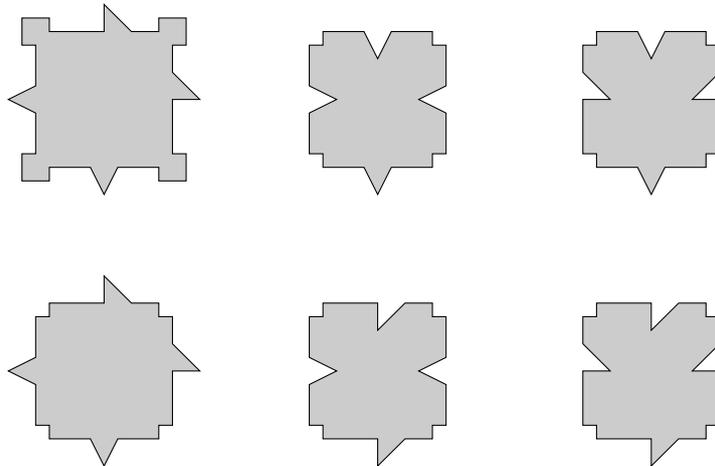
**Exercice 14.** Les sous-shift suivants sont-ils de type fini ou sofique :

1.  $\mathbf{T} = \{x \in \{0, 1\}^{\mathbb{Z}^d} : \text{le symbole } 1 \text{ apparait au plus une fois dans } x\}$ .
2.  $\mathbf{T} = \left\{ x \in \{0, 1, 2\}^{\mathbb{Z}^2} : \begin{array}{l} \text{si } x_{i_0, j_0} = 2 \text{ alors } x_{i_0, j} = 2 \text{ pour tout } j \in \mathbb{Z} \\ \text{pour } i \neq i_0 \text{ on a } x_{i, j} \neq 2 \text{ et } x_{i, j} = x_{i, j_0 - j} \text{ pour tout } j \in \mathbb{Z} \end{array} \right\}$ .
3.  $\mathbf{T} = \{x \in \{0, 1\}^{\mathbb{Z}^2} : \text{les composantes connexes finies de } 1 \text{ de } x \text{ sont paires}\}$  où une composante connexe de 1 de la configuration  $x \in \{0, 1\}^{\mathbb{Z}^2}$  est un ensemble  $\mathbb{U} \subset \mathbb{Z}^2$  tel que  $x_{\mathbb{U}} = 1^{\mathbb{U}}$ , si  $u, u' \in \mathbb{U}$  alors on peut aller de  $u$  à  $u'$  en se déplaçant suivant  $\pm(1, 0)$  ou  $\pm(0, 1)$  sans sortir de  $\mathbb{U}$  et si  $u \notin \mathbb{U}$  et  $\{u + (1, 0), u - (-1, 0), u + (0, 1), u - (0, 1)\} \cap \mathbb{U} \neq \emptyset$  alors  $x_u = 0$ .
4.  $\mathbf{T} = \{x \in \{0, 1\}^{\mathbb{Z}^2} : \text{les composantes connexes finies de } 1 \text{ de } x \text{ sont impaires}\}$ .

## 7.2 Pavage apériodique

**Exercice 15.** Prouver que pour le pavage de Robinson, si on supprime les contraintes supplémentaires qui forcent le réseau alors il existe une configuration périodique.

**Exercice 16.** Montrer que les tuiles suivantes forment nécessairement un pavage apériodique, toute type de rotations sont autorisées (il y a un lien avec Robinson).



## 7.3 Problème de décisions

**Exercice 17.** Déterminer si la machine de Turing  $\mathcal{M} = (\Gamma = \{a, \#\}, \#, Q = \{s, q, p, h\}, s, h, \delta)$  s'arrête où  $\delta$  est défini par :

$$\begin{array}{lll} \delta(s, \#) = (q, a, \rightarrow) & \delta(s, a) = (h, a, \rightarrow) & \delta(q, \#) = (q, a, \leftarrow) \\ \delta(q, a) = (p, \#, \rightarrow) & \delta(p, \#) = (p, a, \leftarrow) & \delta(p, a) = (s, a, \leftarrow) \end{array}$$

**Exercice 18.** Dire si les problèmes de décisions suivants sont décidables, semi-décidable ou indécidable :

1. “Étant donné une machine de Turing  $\mathcal{M}$  et un symbole  $a$ , est-ce que la machine  $\mathcal{M}$  écrit le symbole  $a$  sur son ruban à un moment donné?”
2. “Étant donné une machine de Turing  $\mathcal{M}$  et un état  $q$ , est ce que la machine  $\mathcal{M}$  passe une infinité de fois par  $q$  en partant du ruban vide?”

**Exercice 19.** Pour tout entier  $n$ , on note  $\mathcal{M}(n)$  l'ensemble des machine de Turing qui :

- ont  $n + 1$  états, 1 état d'arrêt et  $n$  de calcul,
- ont un alphabet à deux lettres  $\{1, \#\}$ ,
- s'arrêtent sur l'entrée vide.

Ce sont donc les machines qui s'arrêtent de la forme  $(\Gamma = \{a, \#\}, \#, Q = \{s_1, \dots, s_n, h\}, s_1, h, \delta)$ . On note  $S(n)$  le nombre maximal de pas de calcul effectué par une machine de  $\mathcal{M}(n)$  avant de s'arrêter.

1. Montrer que  $S(2) \geq 6$  et  $S(3) \geq 21$ . En fait il est connu que  $S(2) = 6$ ,  $S(3) = 21$ ,  $S(4) = 107$ ,  $S(5) \geq 47176870$  et  $S(6) \geq 3.10^{2879}$ .
2. Montrer qu'il n'existe pas de programme qui prend en entrée un nombre  $n$  et sort un nombre plus grand que  $S(n)$ . On appelle ce problème “le problème des castors affairés”.

**Exercice 20.** Étant donné une un SFT  $\mathbf{T} = \mathbf{T}(\mathcal{A}, 2, \mathcal{F})$  et une lettre  $a \in \mathcal{A}$  étudier la décidabilité des problèmes suivants :

1. “Est ce qu'il existe une orbite périodique de  $\mathbf{T}$  contenant  $a$ ?”
2. “Est ce que toutes les orbites périodiques de  $\mathbf{T}$  contiennent  $a$ ?”
3. “Est ce qu'il existe une configuration  $x$  et  $\{a, b\} \subset \mathcal{A}$  tel que  $x \in \{a, b\}^{\mathbb{Z}^2}$ ?”

**Exercice 21.** Prouvez que les problèmes de décision suivants prenant en entrée un sous-shift de type fini  $\mathbf{T}$  sont indécidables :

1. “Est ce que  $\mathbf{T} = \emptyset$ ?”
2. “Est ce que  $\mathbf{T}$  est fini?”
3. “Est ce que  $\mathbf{T}$  est transitif?”
4. “Est ce que  $\mathbf{T}$  est minimal?”

# Bibliographie

- [AGS92] Robert Ammann, Branko Grunbaum, and G.C. Shephard. Aperiodic tiles. *Discrete and Computational Geometry*, 8(1) :1–25, December 1992.
- [AS11] Nathalie Aubrun and Mathieu Sablik. Simulation of effective subshifts by two-dimensional subshift of finite type. *To appear in Acta Applicandae Mathematicae, Hang Kim Memorial Volume.*, 2011.
- [Ber66] Robert Berger. *The Undecidability of the Domino Problem*. Number 66 in Memoirs of the American Mathematical Society. The American Mathematical Society, 1966.
- [Bur88] S. E. Burkov. Absence of weak local rules for the planar quasicrystalline tiling with the 8-fold rotational symmetry. *Comm. Math. Phys.*, 119(4) :667–675, 1988.
- [DRS10] Bruno Durand, Andrei Romashchenko, and Alexander Shen. Fixed-point tile sets and their applications. to appear. See also [?], 2010.
- [FBS] N. Pythéas Fogg, Xavier Bressaud, and Mathieu Sablik. Combinatory tilings : Substitutive construction vs matching rules. *En cours*.
- [FO10] Thomas Fernique and Nicolas Ollinger. Combinatorial substitutions and sofic tilings. In Jarkko Kari, editor, *Second Symposium on Cellular Automata "Journéacute;es Automates Cellulaires"*, JAC 2010, Turku, Finland, December 15-17, 2010. *Proceedings*. Turku Center for Computer Science, 2010.
- [FS12] Thomas Fernique and Mathieu Sablik. Local rules for computable planar tilings. *Soumis*, 2012.
- [GK72] Yuri Gurevich and I Koryakov. Remarks on Berger’s paper on the domino problem. *Siberian Math. Journal*, 1972.
- [GS98] Chaim Goodman-Strauss. Matching Rules and Substitution Tilings . *Annals of Mathematics*, 147 :181–223, 1998.
- [Hed69] Gustav Arnold Hedlund. Endomorphisms and automorphisms of the shift dynamical system. *Theory of Computing Systems*, 3(4) :320–375, 1969.
- [HM10] Michael Hochman and Tom Meyerovitch. A characterization of the entropies of multidimensional shifts of finite type. *Annals of Mathematics*, 171(3) :2011–2038, May 2010.
- [Hoc09] Michael Hochman. On the dynamics and recursive properties of multidimensional symbolic systems. *Invent. Math.*, 176(1) :131–167, 2009.
- [Hoo66] Philip K. Hooper. The Undecidability of the Turing Machine Immortality Problem. *Journal of Symbolic Logic*, 31(2) :219–234, June 1966.
- [II96] Karel Culik II. An aperiodic set of 13 Wang tiles. *Discrete Mathematics*, 160 :245–251, 1996.
- [Kar96] Jarkko Kari. A small aperiodic set of Wang tiles. *Discrete Mathematics*, 160 :259–264, 1996.

- [Kar07] Jarkko Kari. The Tiling Problem Revisited . In *Machines, Computations, and Universality (MCU)*, number 4664 in Lecture Notes in Computer Science, pages 72–79, 2007.
- [Le97] Thang T. Q. Le. Local rules for quasiperiodic tilings. In *The mathematics of long-range aperiodic order (Waterloo, ON, 1995)*, volume 489 of *NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci.*, pages 331–366. Kluwer Acad. Publ., Dordrecht, 1997.
- [Lev88] L. S. Levitov. Local rules for quasicrystals. *Comm. Math. Phys.*, 119(4) :627–666, 1988.
- [Lin84] D. A. Lind. The entropies of topological Markov shifts and a related class of algebraic integers. *Ergodic Theory Dynam. Systems*, 4(2) :283–300, 1984.
- [LM95] Douglas A. Lind and Brian Marcus. *An Introduction to Symbolic Dynamics and Coding*. Cambridge University Press, New York, NY, USA, 1995.
- [Moz89] Shahar Mozes. Tilings, substitutions systems and dynamical systems generated by them. *J. d’Analyse Math.*, 53 :139–186, 1989.
- [Pen78] R. Penrose. Pentaplexity : A class of Non-Periodic Tilings of the Plane. *Eureka*, 39, 1978.
- [PS10] Ronie Pavlov and Michael Schraudner. Classification of sofic projective subdynamics of multidimensional shifts of finite type. *Soumis*, 2010.
- [Rob71] Raphael M. Robinson. Undecidability and Nonperiodicity for Tilings of the Plane. *Inventiones Mathematicae*, 12(3), 1971.
- [Soc89] Joshua E. S. Socolar. Simple octagonal and dodecagonal quasicrystals. *Phys. Rev. B* (3), 39(15) :519–551, 1989.
- [Soc90] Joshua E. S. Socolar. Weak matching rules for quasicrystals. *Comm. Math. Phys.*, 129(3) :599–619, 1990.
- [SS12] Mathieu Sablik and Michael Schraudner. Speed of convergence towards an effective subshift. *Soumis*, 2012.
- [Tha95] Le Tu Quoc Thang. Local rules for pentagonal quasi-crystals. *Discrete Comput. Geom.*, 14(1) :31–70, 1995.
- [TPS92] Le Tu Quoc Thang, Sergey Piunikhin, and Vladimir Sadov. Local rules for quasiperiodic tilings of quadratic 2-planes in  $\mathbf{R}^4$ . *Comm. Math. Phys.*, 150(1) :23–44, 1992.
- [Wan61] Hao Wang. Proving theorems by Pattern Recognition II. *Bell Systems technical journal*, 40 :1–41, 1961.
- [Wei73] Benjamin Weiss. Subshifts of finite type and sofic systems. *Monatsh. Math.*, 77 :462–474, 1973.